



UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
VERITATE SOLA NOBIS IMPONETUR VIRILISTOGA. 1948

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica

TRABAJO DE DIPLOMA

Aplicaciones avanzadas con microcontroladores PICs.

Autores: Saeddy Castro Ávila.

Halder Vladimir Galdo Navarro.

Tutor: Msc. Carlos Alberto Bazán Prieto.

Santa Clara

2007

"Año 49 de la Revolución"



Universidad Central "Marta Abreu" de Las Villas
Facultad de Ingeniería Eléctrica
Departamento de Telecomunicaciones y Electrónica



TRABAJO DE DIPLOMA

Aplicaciones avanzadas con microcontroladores PICs.

Autor: Saeddy Castro Ávila.

E-mail: saeddy@uclv.edu.cu

Halder Vladimir Galdo Navarro.

E-mail: halder_vladimir@uclv.edu.cu

Tutor: M. Sc. Carlos Alberto Bazán Prieto.

Profesor, Dpto de Electrónica y Telecomunicaciones
Facultad de Ing. Eléctrica. UCLV.

E-mail: cabazan@uclv.edu.cu

Santa Clara

2007

"Año 49 de la Revolución"



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Telecomunicaciones y Electrónica, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Autor

Firma del Jefe de
Departamento donde se
defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

PENSAMIENTO

“Nuestro mayor orgullo no ha de ser el no haber fracasado nunca, sino en cambio, el habernos puesto de pie cada vez que hayamos caído.”

Confucio

DEDICATORIA

En primer lugar a Dios, al cual le debemos todo lo que somos y todo lo que hemos logrado en la vida.

A mis padres, hermano, esposa e hijo, que tanto se han sacrificado para que este sueño se haga realidad.

** * **

A Indira, por su apoyo incondicional.

A mis padres, que constituyen un ejemplo a seguir.

A mis abuelos Julia y Vicente, mis fieles ángeles guardianes.

AGRADECIMIENTOS

Al colectivo docente de la Facultad de Ingeniería Eléctrica de la Universidad Central “Marta Abreu” de Las Villas, por haber puesto todo su empeño en hacer de nosotros excelentes profesionales.

A nuestro tutor Carlos Alberto Bazán, por la ayuda y el apoyo brindados.

A Mario y Migdalia, nuestros padres en este proyecto, sin ellos este sueño no se hubiera materializado.

Al ingeniero Alleini Machado por sus útiles consejos.

* * *

Al Departamento de Mantenimiento del Hotel Brisas Santa Lucía.

A todos los compañeros que nos atendieron durante nuestras prácticas laborales.

A la sucursal de Copextel S.A en Santa Lucía.

A mis amigos Julio, José y Cristina en Santa Clara.

A mis amigos Yasiel, Maikel, Yandy, Yanela, Nayibis y Leosvanis.

* * *

A mis grandes hermanos de batalla: Rubén, Erisdel y Raulier.

A mi tía Lolita y a Yideliza y familia, mi retaguardia segura en Santa Clara.

Al resto de nuestros familiares y amigos, no nos alcanza el espacio para mencionarlos a todos pero los hemos tenido siempre presentes.

A todos, simplemente gracias.

TAREA TÉCNICA

1. Revisión crítica de la bibliografía del tema y análisis de estado del arte.
2. Estudio de subsistemas avanzados en los PICs.
3. Elaborar aplicaciones didácticas empleando los subsistemas anteriores.
4. Integrar diferentes subsistemas en una aplicación.
5. Confección y presentación del informe.

Firma del Autor

Firma del Tutor

RESUMEN

En el presente trabajo se muestra un análisis de algunos interfaces presentes en los microcontroladores PICs, particularmente el Interfaz de Comunicación Serie (SCI) y el Interfaz de Conversión Análogo Digital (CAD). Se describen ejemplos de aplicaciones utilizando estos interfaces, las cuales tienen un alto valor docente, mencionándose sus características de software y de hardware. Finalmente se realiza la integración de los interfaces estudiados en una aplicación más compleja: un cartel pasa-mensajes, el cual se utilizará como elemento decorativo e informativo para la Facultad de Ingeniería Eléctrica. Este servirá además de base a posteriores aportes a la docencia.

Palabras claves: Microcontroladores PICs, Interfaz de Comunicación Serie (SCI), Interfaz de Conversión Análogo Digital (CAD), cartel pasa-mensajes.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTO	iii
TAREA TÉCNICA	iiv
RESUMEN	v
INTRODUCCIÓN	1
Organización del informe	2
CAPÍTULO 1. Los microcontroladores PICs. El PIC 16F877A.	4
1.1 Características de los microcontroladores PIC.	4
1.1.1 Las familias de microcontroladores PIC.	6
1.1.2 Elección de un PIC para una aplicación específica.	9
1.2 El microcontrolador 16F877A.	10
1.2.1 Principales características del 16F877A.	10
1.3 Subsistemas avanzados en el PIC 16F877A.	12
1.3.1 Interfaz de comunicación serie (SCI).	12
1.3.2 Interfaz de conversión análogo digital (CAD).	20
CAPÍTULO 2. Aplicaciones con subsistemas avanzados del PIC 16F877A.	28
2.1 Comunicación serie PC-PIC con visualización a través de display LCD.	28

2.1.1	Esquema circuital.....	29
2.1.2	Diagrama en bloques y descripción del software.	30
2.2	Medición de temperatura con visualización a través de display LCD.	32
2.2.1	Esquema circuital.....	33
2.2.2	Diagrama en bloques y descripción del software.	34
2.3	Comunicación teclado-PIC con visualización a través del display LCD.	38
2.3.1	Esquema circuital.....	39
2.3.2	Diagrama en bloques y descripción del software	41
CAPÍTULO 3. Diseño, programación y montaje del cartel pasa-mensajes.		44
3.1	Generalidades.	44
3.2	Diseño y análisis del Hardware.	45
3.2.1	Características generales.....	45
3.2.2	Matrices de LEDs.....	46
3.2.3	Registros de desplazamiento.....	47
3.2.4	Fuente de alimentación.....	51
3.2.5	Montaje y pruebas.	51
3.3	Algoritmos y programas.	52
3.3.1	Diagrama de bloques de las principales subrutinas y del programa principal.	53
3.3.2	Comunicación PC-cartel pasa-mensajes.....	62
3.4	Análisis económico.	63
3.5	Manual de usuario.	64
CONCLUSIONES Y RECOMENDACIONES		66
Conclusiones		66
Recomendaciones		66

REFERENCIAS BIBLIOGRÁFICAS.....	67
ANEXOS.....	70
Anexo I. Comunicación serie PC-PIC con visualización a través de display LCD..	70
Anexo II. Medición de temperatura con visualización a través de display LCD.	70
Anexo III. Comunicación teclado-PIC con visualización a través del display LCD.	71
Anexo IV. Valores de SCANCODE para los caracteres de teclado.....	72
Anexo V. Código ASCII para caracteres de teclado.....	73
Anexo VI. Circuito impreso del cartel pasa-mensajes.....	74
Anexo VII. El cartel pasa-mensajes.....	74

INTRODUCCIÓN

El microcontrolador es uno de los logros más sobresalientes de la evolución tecnológica que ha experimentado la humanidad. Cada año que pasa se acerca más al centro de nuestras vidas, forjándose un sitio en el núcleo de una máquina tras otra. Evolucionando a mayor velocidad que cualquier otro invento, la capacidad del microcontrolador ha aumentado 10.000 veces en los últimos 25 años. Su presencia ha comenzado a cambiar la forma en que percibimos el mundo e incluso a nosotros mismos. Cada vez se hace más difícil pasar por alto el microcontrolador como otro simple producto en una larga línea de innovaciones tecnológicas.

Ninguna otra invención en la historia se ha diseminado tan aprisa por todo el mundo o ha tocado tan profundamente tantos aspectos de la existencia humana. Hoy existen millones de microchips de alguna clase en uso y el número de productos que funcionan en base a uno o varios microcontroladores aumenta de forma exponencial. El mayor atributo del microcontrolador es que puede integrar inteligencia casi a cualquier artefacto. Se le puede entrenar para adaptarse a su entorno, responder a condiciones cambiantes y volverse más eficiente y que responda a las necesidades únicas de sus usuarios.

La presencia de subsistemas avanzados capaces de dotar a los microcontroladores de prestaciones cada vez más valiosas hace necesario el dominio y empleo de estos subsistemas en el desarrollo de aplicaciones de creciente complejidad.

De lo anteriormente expuesto se desprenden las siguientes interrogantes científicas:

¿Cómo se podría llevar a cabo, en el desarrollo de una aplicación, la integración de diferentes subsistemas avanzados de los PICs?

¿Cómo desarrollar el ensamblaje, compilación y puesta a punto de programas utilizando herramientas de software avanzadas?

Los objetivos del presente trabajo son la consecuencia directa de las interrogantes científicas. Entre ellos están, realizar un estudio profundo de algunos de los subsistemas avanzados de los microcontroladores hasta dominar su funcionamiento. Emplear estos subsistemas en el desarrollo de aplicaciones sencillas que nos sirvan de base para desarrollar una aplicación que integre estos subsistemas y utilizar diferentes herramientas de software avanzadas para el ensamblaje, compilación, puesta a punto de las aplicaciones y que además nos permitan realizar proceso de grabado de los PICs.

Para llevar a cabo estos objetivos se ejecutaron las siguientes tareas: primero, búsqueda bibliográfica y estudio de algunos de los subsistemas avanzados presentes en los microcontroladores PICs; segundo, empleo de estos subsistemas en pequeñas aplicaciones, sencillas pero de gran valor didáctico; tercero, integración de los subsistemas en una aplicación de mayor complejidad; cuarto, confección, depuración y puesta a punto de los diferentes programas haciendo uso de diferentes herramientas de software.

Organización del informe

Este informe se ha estructurado en tres capítulos que tratan las siguientes temáticas:

CAPITULO 1: Se aborda lo básico referente a las características generales de Microcontroladores PICs, haciendo énfasis particularmente en el PIC 16F877A y

sus subsistemas avanzados de los que haremos uso para desarrollar aplicaciones.

CAPITULO 2: Se describen las aplicaciones obtenidas del trabajo con los subsistemas avanzados de los PICs, mencionando sus características tanto de software como de hardware.

CAPITULO 3: Este capítulo estará dedicado a la principal aplicación del presente trabajo: el cartel pasa-mensajes, la cual integra los subsistemas avanzados estudiados. De esta aplicación se describe tanto el hardware como el software. Además se dedica un epígrafe a la descripción del software utilizado para la comunicación con el cartel pasa-mensajes: el Hyper Terminal.

CAPÍTULO 1. Los microcontroladores PICs. El PIC 16F877A.

En este capítulo se abordan las características generales de los microcontroladores PICs, sus prestaciones fundamentales y sus principales productores. Se describen las distintas familias de PIC que se fabrican y comercializan en el mercado mundial y los aspectos fundamentales a tener en cuenta a la hora de decidirnos por un determinado PIC para emplearlo en alguna aplicación específica.

Dentro de la amplia gama de microcontroladores PIC se enfatiza en el 16F877A, que es el que se empleará para las aplicaciones de nuestro trabajo. Se describen sus principales características y se tratan dos de sus subsistemas avanzados utilizados: el Interfaz de Comunicación Serie (SCI) y el Interfaz de Conversión Análogo Digital.

1.1 Características de los microcontroladores PIC.

Los microcontroladores PICs son fabricados por una gran variedad de productores a nivel mundial, destacándose cuatro compañías: National, Philips, Motorola y Microchip. Esta última es la más prestigiosa en este renglón. Las características de los PICs producidos por estos gigantes son muy similares así como sus prestaciones. (Segredo, 2006)

Una gran cantidad de centros educacionales especializados en ingeniería electrónica, automática o de telecomunicaciones se están dedicando el estudio de los microcontroladores PIC. Ejemplos de ello son la Escuela Superior de Ingenieros de Sevilla (España), la Universidad de Kent (Reino Unido) y la Universidad de Oxford (Reino Unido) entre otros. Además de clases presenciales,

se publican libros con autoría de profesores, hay presencia en Internet abundante información sobre el tema y existe un gran respaldo técnico por parte de los fabricantes en su mayoría disponible de forma gratuita. Todos estos elementos han conllevado a la popularidad de los microcontroladores PICs y al interés cada vez más creciente en ellos por parte de los seguidores de los últimos avances tecnológicos. (Segredo, 2006)

Características que han popularizado el uso de los PICs:

1. Su costo es comparativamente inferior al de sus competidores.
2. Poseen una elevada velocidad de funcionamiento.
3. Tienen un juego reducido de instrucciones; 35 en la gama media.
4. Los programas son compactos.
5. Bajo consumo unido a un amplio rango de voltajes de alimentación.
6. Muchas herramientas de software se pueden recoger libremente a través de INTERNET desde Microchip.
7. Existen una gran variedad de herramientas de hardware que permiten grabar, depurar, borrar y comprobar el comportamiento de los PIC.
8. La gran variedad de modelo de los PIC permite elegir el que mejor responde a los requerimientos de la aplicación.

Los PICs a pesar de su reciente aparición ocupan posiciones de primera línea en el ranking mundial, existiendo una gran competencia con otros microcontroladores producidos por gigantes como Intel y Motorola. En 1990 ocupaban el puesto vigésimo y actualmente son uno de los más vendidos en el mundo cada año. (Angulo y Angulo, 1999)

Los PICs actuales vienen con una amplia gama de mejoras de hardware incorporadas:

1. Núcleos de UCP de 8/16 bits con Arquitectura Harvard modificada.
2. Memoria Flash y ROM disponible desde 256 bytes a 256 Kbytes.

3. Puertos de E/S (típicamente 0 a 5,5 voltios).
4. Temporizadores de 8/16 bits.
5. Tecnología Nanowatt para modos de control de energía.
6. Periféricos serie sincrónicos y asincrónicos: USART, AUSART, EUSART.
7. Conversores analógico/digital de 10-12 bits.
8. Comparadores de tensión.
9. Módulos de captura y comparación PWM.
10. Controladores LCD.
11. Periférico MSSP para comunicaciones I2C, SPI, y I2S.
12. Memoria EEPROM interna con duración de hasta un millón de ciclos de lectura/escritura.
13. Periféricos de control de motores.
14. Soporte de interfaz USB.
15. Soporte de controlador Ethernet.
16. Soporte de controlador Irda.
17. Otros.

(Wikipedia, 2007)

1.1.1 Las familias de microcontroladores PIC.

Para resolver aplicaciones sencillas se precisan pocos recursos, en cambio, las aplicaciones grandes requieren numerosos y potentes. Siguiendo esta filosofía Microchip construye diversos modelos de microcontroladores orientados a cubrir, de forma óptima, las necesidades de cada proyecto. Así, hay disponibles microcontroladores sencillos y baratos para atender las aplicaciones simples y otros complejos y más costosos para las de mucha envergadura. Microchip dispone de cuatro familias de microcontroladores de 8 bits para adaptarse a las necesidades de la mayoría de los clientes potenciales. (Sánchez, 2005).

GAMA DE MICROCONTROLADORES PEQUEÑOS: PIC12C(F)XXX DE 8 PINES.

Se trata de un grupo de PIC de reciente aparición que ha acaparado la atención del mercado. Su principal característica es su reducido tamaño al disponer todos sus componentes de 8 pines. Se alimenta con un voltaje de corriente continua

comprendido entre 2.5V y 5V y consumen menos de 2mA cuando trabajan a 5V y 4 MHz. El formato de sus instrucciones puede ser de 12 o 14 bits y su repertorio es de 33 o 35 instrucciones respectivamente. Aunque los PIC pequeños solo tienen 8 pines, pueden destinar hasta 6 como líneas de Entrada / Salida para lo periféricos porque disponen de un oscilador interno RC.

GAMA BAJA O BASICA: PIC16C5X CON INSTRUCCIONES DE 12 BITS.

Se trata de una serie de PIC con recursos limitados, pero con una de las mejores relaciones costos prestaciones. Sus versiones están encapsuladas con 18 y 28 pines y pueden alimentarse a partir de una tensión de 2.5V, lo que les hace ideales en las aplicaciones que funcionan con pilas. Tienen un repertorio de 33 instrucciones cuyo formato consta de 12 bits no admiten ningún tipo de interrupción y la pila solo dispone de 2 niveles.

GAMA MEDIA. PIC16CXXX CON INSTRUCCIONES DE 14 BITS.

Es la gama mas variada y completa de los PIC. Abarca modelos con encapsulados de 18 pines hasta 68, cubriendo varias opciones que integran abundantes periféricos. El repertorio de instrucciones es de 35 de 14 bits cada una y compatible con el de la gama baja. Sus distintos modelos contienen todos los recursos que se precisan en las aplicaciones de lo microcontroladores de 8 bits. También disponen de interrupciones y una pila de 8 niveles que permiten el anidamiento de subrutinas.

GAMA ALTA: PIC 17CXXX CON INSTRUCCIONES DE 16 BITS.

Se alcanzan las 58 instrucciones de 16 bits en el repertorio y sus modelos disponen de un sistema de gestión de interrupciones vectorizadas muy potentes. También incluyen variados controladores de periféricos, puertos de comunicación serie y paralelo con elementos externos y multiplicador hardware de gran velocidad. La característica más destacable de los componentes de esta gama es su arquitectura abierta, que consiste en la posibilidad de ampliación del microcontrolador con elementos externos. Para este fin, los pines sacan al exterior las líneas de buses de datos, direcciones y control, a las que se conectan memorias o controladores de periféricos. Esta filosofía de construcción

de sistema es la que se empleaba en los microprocesadores y no suele ser una práctica habitual cuando se emplean microcontroladores.

Además existen algunos PICs de reciente aparición en el mercado y que incluyen subsistemas muy avanzados.

PICs WIRELESS (rfPIC).

El microcontrolador rfPIC integra todas las prestaciones del PICmicro de Microchip Technology Inc. con la capacidad de comunicación wireless UHF para aplicaciones RF de baja potencia. Estos dispositivos ofrecen un diseño muy comprimido para ajustarse a los cada vez más demandados requerimientos de miniaturización en aparatos electrónicos. (Wikipedia, 2007)

PICs PARA PROCESAMIENTO DIGITAL DE SEÑALES. (dsPIC).

Los dsPIC son el último lanzamiento de Microchip, comenzando a producirlos a gran escala a finales de 2004. Son los primeros PICs con bus de datos inherente de 16 bits. Incorporan todas las posibilidades de los anteriores PICs y añaden varias operaciones de DSP implementadas en hardware, como multiplicación con suma de acumulador (multiply-accumulate, o MAC), barrel shifting, bit reversion o multiplicación 16x16 bits. (Wikipedia, 2007).

PICs CON PUERTO USB.

La mayoría de microcontroladores USB están optimizados para aplicaciones del mercado de consumo y del PC más que para diseños industriales. Las nuevas familias de microcontroladores PIC USB de Microchip Technology, Inc. ofrecen los beneficios del USB 'Full Speed' para un amplio rango de aplicaciones que pueden operar en entornos duros, industriales y conectados a ordenadores personales. El objetivo de las aplicaciones para estos productos incluye el mercado del automóvil, aplicaciones industriales, médicas así como aplicaciones de consumo avanzadas y otros dispositivos conectados al PC o alimentados por baterías. (Wikipedia, 2007).

ETHERNET PICs.

También la Ethernet ha sido conquistada por los microcontroladores PIC. Ejemplo de ello es el PIC18F452 de la Microchip Technology, Inc. Este tipo de PIC es generalmente incorporado en una tarjeta con memoria externa y conectores RS232 u otros. Sus aplicaciones típicas incluyen Servidores de Web http, clientes de correo, conversión de Ethernet a interfaz RS232, control remoto por servidor de Web entre otras. (Wikipedia, 2007)

1.1.2 Elección de un PIC para una aplicación específica.

Es difícil escoger el microcontrolador a emplear en un diseño concreto, dada la gran variedad existente en el mundo. En realidad no se puede hablar del mejor microcontrolador, en cada aplicación son las características específicas quienes determinan el más conveniente. Las características fundamentales que nos permiten escoger una u otra opción son:

1. Sencillez de manejo.
2. Buena información.
3. Precio.
4. Buen promedio de parámetros:
 - Velocidad.
 - Consumo.
 - Tamaño.
 - Alimentación.
 - Código compacto.
5. Herramientas de desarrollo fáciles y baratas.
6. Diseño rápido.

(Angulo y Angulo, 1999).

También podemos manejar otros criterios entre los que tenemos:

1. Longitud de palabra: 4 bits es apropiado para algunas aplicaciones pero lo limitan consideraciones de costo, 8 bits resulta satisfactorio en casi todos los casos en tanto que 16 y 32 bits se reservan para aplicaciones con gran necesidad de direccionamiento.
2. Periféricos: El diseñador debe conocer las operaciones necesarias y si estas estarán dentro o fuera del microcontrolador, la cantidad de bits necesarios en el periférico es quien decide los bits en el microcontrolador.
3. Costo: Los microcontroladores de 4 bits son los más baratos, la mejor relación costo/funcionalidad la tienen los de 8 bits, la tecnología de menor consumo es la CMOS pero es un poco más cara que la NMOS. (Bazán, 2006).

1.2 El microcontrolador 16F877A.

De acuerdo a las características de las aplicaciones que vamos a desarrollar y teniendo en cuenta las condiciones de disponibilidad presentes en nuestra situación nos decidimos por el empleo de un PIC de la gama media y de los disponibles de esta familia escogimos el microcontrolador 16F877A debido fundamentalmente a que este PIC posee los subsistemas avanzadas de comunicación serie (SCI) y de conversión análogo-digital (CAD), además posee 40 terminales lo cual nos da una amplia posibilidad de interacción con diversos periféricos.

1.2.1 Principales características del 16F877A.

El modelo 16F877A posee varias características que hacen a este microcontrolador un dispositivo muy versátil, eficiente y práctico para ser empleado en diversas aplicaciones que posteriormente serán detalladas. El diagrama de pines de este microcontrolador se muestra en la figura 1.1.(Herrera y Hernández, 2004)

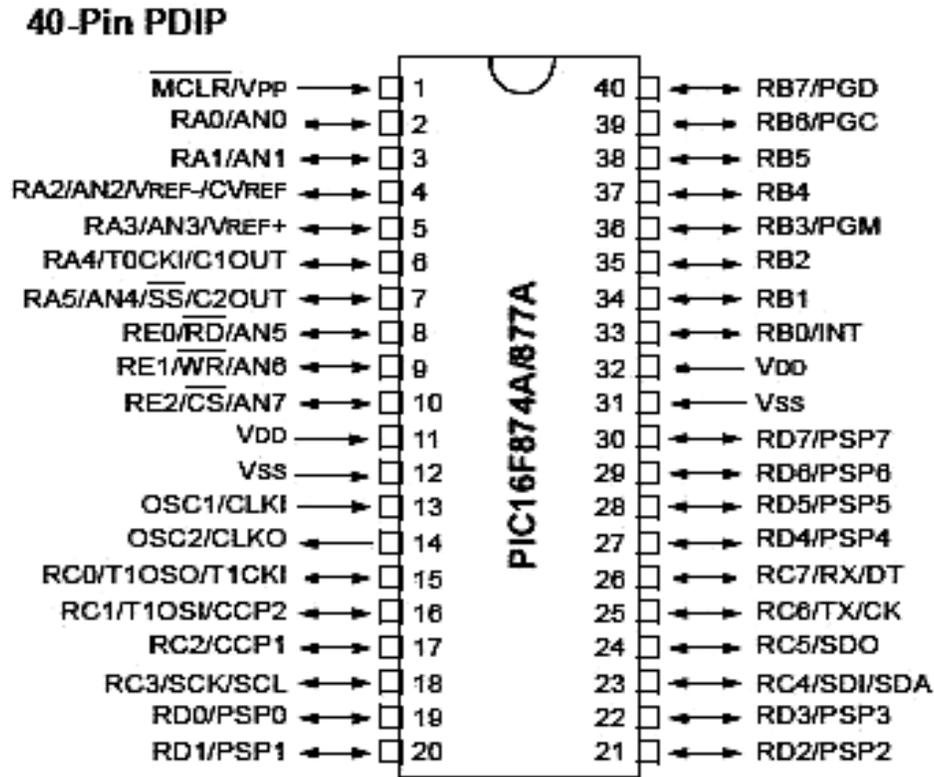


Figura 1.1. Diagrama de pines para el 16F877A.

Características:

1. Memoria de Programa tipo Flash 8Kx14.
2. Memoria Datos 368 bytes.
3. EEPROM 256 bytes.
4. 33 pines de Entrada/Salida.
5. Encapsulado: 40 pines DIP, 44 pines PLCC y 44 pines TQFP.
6. Soporta Xtal 20MHz.
7. Voltaje de Operación: 2.0 hasta 5.5VDC. (Microchip, 2003b).

Periféricos:

1. 1 Conversor A/D de 10-bits (8 canales).
2. 2 Módulos CCP (Captura, Comparador, PWM).
3. 1 Modulo I²C.
4. 1 USART (Puerto Serie).

5. 2 Timers de 8 bits.
6. 1 Timer 16 bits. (Microchip, 2003b).

1.3 Subsistemas avanzados en el PIC 16F877A.

1.3.1 Interfaz de comunicación serie (SCI).

La interfaz de comunicación serie se conoce como Universal Synchronous Asynchronous Receiver Transmitter (USART) o de manera más sencilla como Serial Communication Interface (SCI). El SCI puede ser configurado como un sistema asincrónico bidireccional que puede comunicarse con equipos terminales de datos como las computadoras personales o puede ser configurado como un sistema sincrónico unidireccional que permite la comunicación con periféricos como conversores análogo-digitales o digito-analógicos, y otros circuitos integrados. (Microchip, 2003b)

El SCI puede ser configurado en los siguientes modos:

- Asincrónico – Bidireccional.
- Sincrónico –Unidireccional Master.
- Sincrónico –Unidireccional Esclavo.

Para utilizar el SCI se deben configurar los terminales RC6/TX/CK y RC7/RX/DT como salida y entrada respectivamente para que puedan ser usados como terminales de TX y RX respectivamente. Para ello se colocan los bits 6 y 7 del registro TRISC en '1' y en '0' respectivamente.

Los registros que permiten habilitar y controlar el estado tanto de la transmisión como de la recepción se describen a continuación.

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

Figura 1.2. Registro de control y estado de la transmisión (TXSTA).

En el PIC 16F877A el registro TXSTA se encuentra en la dirección 98H en el banco 1 dentro del mapa de memoria y sus bits tienen las siguientes funciones:

- Bit 7: bit de selección de la fuente de reloj.
 - En modo asincrónico: se ignora
 - En modo síncronico:
 - 1= modo master (reloj generado internamente).
 - 0= modo esclavo (reloj proveniente de una fuente externa).
- Bit 6: bit de habilitación de la transmisión de 9bits.
 - 1= selecciona la transmisión de 9bits.
 - 0= selecciona la transmisión de 8 bits.
- Bit 5: bit de habilitación de la transmisión.
 - 1= transmisión habilitada.
 - 0= transmisión deshabilitada.
- Bit 4: bit selección del modo de trabajo del SCI.
 - 1= se selecciona el modo síncronico.
 - 0= se selecciona el modo asincrónico.
- Bit 3: no implementado, se lee como un '0'.
- Bit 2: bit de selección de la velocidad de comunicación (Baud Rate).
 - Modo asincronito:
 - 1= alta velocidad.
 - 0= baja velocidad.
 - Modo síncronico: se ignora.
- Bit 1: bit de estado del registro de transmisión (Transmit Shift Register).
 - 1= TSR vacío.
 - 0= TSR lleno.
- Bit 0: constituye el bit 9 de la transmisión de 9 bits en caso de que esta haya sido seleccionada. En caso contrario puede constituir el bit de paridad.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Figura 1.3. Registro de control y estado de la recepción (RCSTA).

En PIC 16F877A el registro RCSTA se encuentra en la dirección 18H dentro del banco 0 del mapa de memoria y sus bits tienen las siguientes funciones:

- Bit 7: bit de habilitación del interfaz serie.
 - 1= interfaz serie habilitada (configura los terminales RC6/TX/CK y RC7/RX/DT como terminales de TX y RX).
 - 0= interfaz serie deshabilitada.
- Bit 6: bit de habilitación de la recepción de 9bits.
 - 1= selecciona la recepción de 9bits.
 - 0= selecciona la recepción de 8 bits.
- Bit 5: bit de habilitación de la recepción simple.
 - Modo asincrónico: se ignora
 - Modo síncrono- master:
 - 1= recepción simple habilitada.
 - 0= recepción simple deshabilitada.
 - Modo síncrono-esclavo: se ignora
- Bit 4: bit de habilitación de la recepción continua
 - Modo asincrónico:
 - 1= recepción continua habilitada.
 - 0= recepción continua deshabilitada.
 - Modo síncrono:
 - 1= recepción continua habilitada hasta que el bit CREN se ponga a '0' (SREN toma el valor de CREN).
 - 0= recepción continua deshabilitada.
- Bit 3: bit de habilitación de la detección de direcciones.
 - Solo usado en modo asincrónico y recepción de 9 bits (RX9='1')
 - 1= habilita la detección de direcciones y habilita la interrupción y carga del buffer de recepción
 - 0= deshabilita la detección de direcciones, el bit 9 puede ser usado como bit de paridad
- Bit 2: bit de error de trama.
 - 1= ocurrió un error de trama.

- 0= no ocurrió ningún error de trama.
- Bit 1: bit de estado del registro de transmisión (Transmit Shift Register).
 - 1= TSR vacío.
 - 0= TSR lleno.
- Bit 0: constituye el bit 9 de la transmisión de 9 bits en caso de que esta haya sido seleccionada. Puede constituir el bit de paridad.

EL GENERADOR DE LA RAZÓN DE BAUD PARA ELSCI (BRG).

El generador de la razón de Baud (BRG) está soportado sobre la transmisión tanto sincrónica como asincrónica. Consiste en un timer dedicado de 8 bit en corrida libre cuyo periodo es controlado por el registro SPBRG. En modo asincrónico el bit BRGH (TXSTA, 2) también controla la razón de Baud. Teniendo el valor deseado de razón de Baud y la frecuencia del oscilador se puede calcular el valor entero más próximo que se le debe asignar al registro SPBRG. La Tabla 1.1 muestra la fórmula para realizar este cálculo.

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64 (X + 1))$	Baud Rate = $F_{osc}/(16 (X + 1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4 (X + 1))$	N/A

Tabla 1.1. Fórmula para obtención de la razón a partir del valor cargado en SPBRG o viceversa, con X = valor de SPBRG (0 a 255).

Cargar un nuevo valor en el registro SPBRG hace que el generador de la razón de Baud se limpie automáticamente. Esto asegura que el timer no espere un desborde para establecer la nueva razón de Baud en la comunicación.

La Tabla 1.2 muestra una tabla con los valores de razón de Baud para sus correspondientes valores cargados en el registro SPBRG y para una frecuencia de oscilador de 4MHz. El error cometido al aproximar el valor de SPBRG al entero más próximo se incluye en la tabla.

BAUD RATE (K)	FOSC = 4 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207
1.2	1.202	0.17	51
2.4	2.404	0.17	25
9.6	8.929	6.99	6
19.2	20.833	8.51	2
28.8	31.250	8.51	1
33.6	-	-	-
57.6	62.500	8.51	0
HIGH	0.244	-	255
LOW	62.500	-	0

Tabla 1.2. Razón de Baud para distintos valores del registro SPBRG y frecuencia de oscilador de 4MHz.

EL SCI EN MODO ASINCRÓNICO.

En modo asincrónico el SCI utiliza el formato estándar NRZ (not returned to zero) el cual consiste en un bit de arranque, ocho o nueve bites de datos y un bit de parada. El formato de datos más común consta de un bit de arranque, ocho bits de datos, un bit de paridad y un bit de parada. El BRG se dedica para establecer una razón de Baud a partir de la frecuencia del oscilador y el valor del registro SPBRG. El SCI transmite o recibe el bit menos significativo (LSB) primero. La transmisión y la recepción son independientes pero usan el mismo formato de datos y la misma razón de Baud. El formato de 9 bits de datos no es soportado por hardware pero puede ser implementado por software. El modo asincrónico es seleccionado poniendo a '0' el bit SYNC (TXSTA, 4).

TRANSMISIÓN EN EL SCI.

El bloque de transmisión para el SCI se muestra en la Figura 1.4.

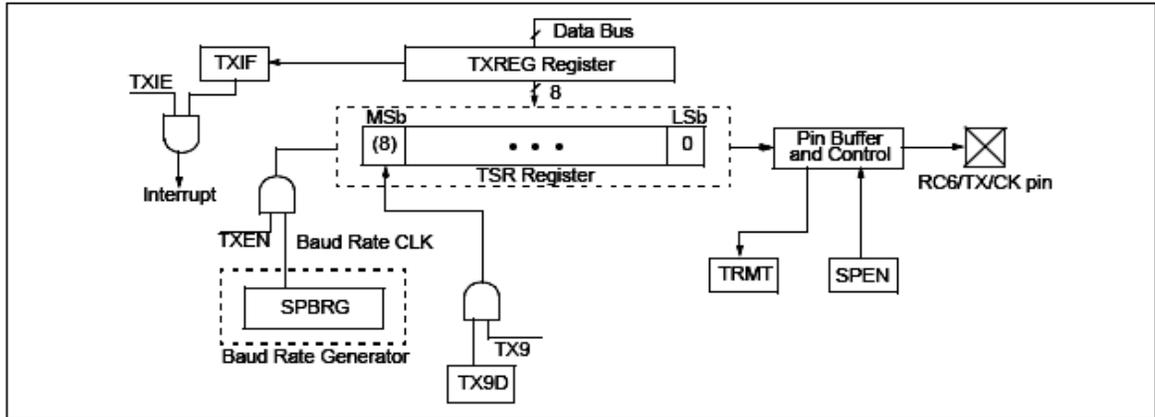


Figura 1.4. Bloque de transmisión del SCI.

El corazón del sistema de transmisión es el registro de transmisión Transmission Shift Register (TSR) el cual se actualiza a partir del buffer de transmisión (registro TXREG). El buffer se carga con un dato por software. El TSR no se actualiza hasta que el bit de arranque del dato actual no es enviado. Inmediatamente después el TSR toma un nuevo valor leyendo de buffer TXREG. La secuencia de transmisión se muestra en la Figura 1.5.

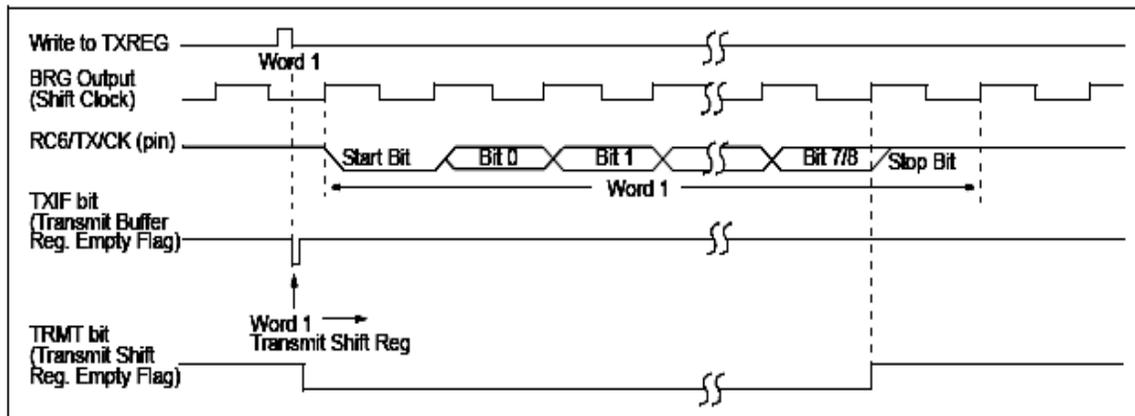


Figura 1.5. Secuencia de transmisión.

Cuando el buffer TXREG transfiere un dato al TSR y queda vacío se activa la bandera TXIF (PIR, 4). Esto ocurre en el tiempo T_{CY} . La interrupción de transmisión se puede habilitar mediante el bit TXIE (PIE, 4). El valor del bit TXIF es independiente del valor del bit TXIE.

Mientras que el bit TXIF indica el estado del buffer TXREG el bit TRMT (TXSTA, 1) muestra el estado del TSR. El bit TMRT solo puede ser leído y es puesto a '1' cuando el TSR esta vacío. El registro TSR no esta en el mapa de memoria y por tanto no es accesible al usuario.

Para realizar una transmisión asincrónica se deben seguir los siguientes pasos:

1. Inicializar el registro SPBRG para una apropiada razón de Baud.
2. Habilitar el puerto serie en modo asincrónico poniendo a '0' el bit SYNC (TXSTA, 4) y a '1' el bit SPEN (RCSTA, 7).
3. Si se quiere habilitar la interrupción poner a '1' el bit TXIE (PIE1, 4).
4. Si se desea usar formato de 9 bits de datos, poner a '1' el bit TX9 (TXSTA, 6).
5. Habilitar la transmisión poniendo a '1' el bit TXEN (TXSTA, 5).
6. Si se selecciono el formato de 9 bits de datos el bit 9 debe ser cargado en el bit TX9D (TXSTA, 0).
7. Cargar dato en el buffer TXREG.
8. Si se esta usando la interrupción asegurarse de que los bits GIE y PEIE (INTCON 7, 6) están puestos a '1'.

RECEPCIÓN EN EL SCI.

El bloque de recepción se muestra en la Figura 1.6.

El corazón del sistema de recepción es el registro de recepción Reception Shift Register (RSR). Después de recibido el bit de parada el valor de RSR se transfiere al registro RCREG si se encuentra vacío. Cuando la transferencia se completa el bit RCIF (PIR1, 5) se pone a '1'. La interrupción puede ser habilitada a través del bit RCIE (PIE1, 5). Si el bit de parada es un '0' entonces se activa el bit de error de trama (RCSTA, 2).

La secuencia de recepción se muestra en la Figura 1.7.

Para efectuar una recepción en modo asincrónico se deben seguir los siguientes pasos:

1. Inicializar el registro SPBRG para una apropiada razón de Baud.

2. Habilitar el puerto serie en modo asincrónico poniendo a '0' el bit SYNC (TXSTA, 4) y a '1' el bit SPEN (RCSTA, 7).
3. Si se quiere habilitar la interrupción poner a '1' el bit RCIE (PIE1, 5).
4. Si se desea usar formato de 9 bits de datos, poner a '1' el bit TX9 (TXSTA, 6).
5. Habilitar la recepción poniendo a '1' el bit CREN (RCSTA, 4).
6. Cuando la recepción se completa se activa RCIF (PIR1, 5) y se genera la interrupción si el bit RCIE (PIE1, 5) fue puesto a '1'.
7. Leer el registro RCSTA para obtener el bit9 de datos (en caso de ser habilitado) y determinar si ocurrió algún error durante el proceso.
8. Leer el dato recibido del registro RCREG.
9. Si ocurrió algún error, eliminarlo poniendo a '0' el bit CREN (RCSTA, 4).
10. Si se está usando la interrupción asegurarse de que los bits GIE y PEIE (INTCON 7, 6) están puestos a '1'.

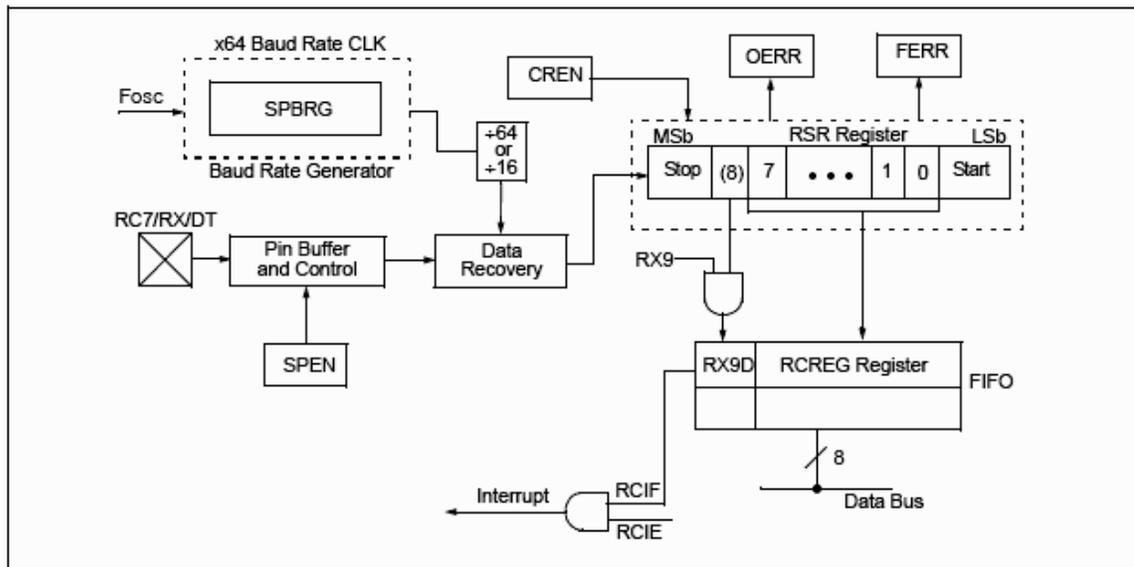


Figura 1.6. Bloque de recepción para el SCI.

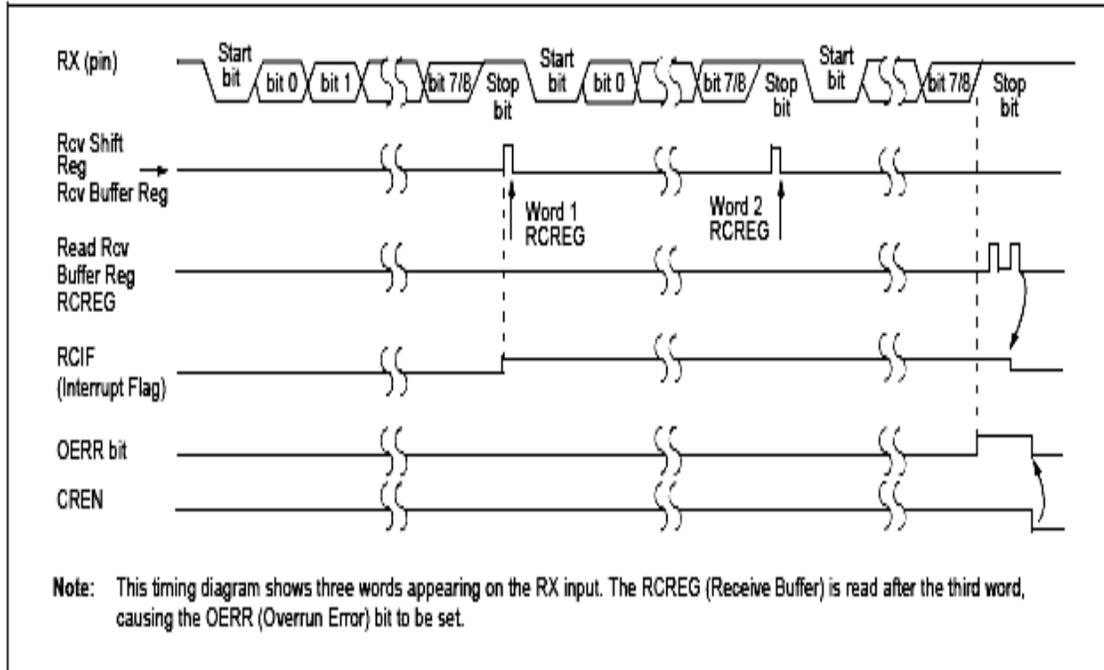


Figura 1.7. Secuencia de transmisión para el SCI.

1.3.2 Interfaz de conversión análogo digital (CAD).

El módulo de conversión análogo-digital (CAD) posee 5 entradas para pastillas de 28 terminales y 8 entradas para el resto. La entrada analógica pasa por un circuito de muestreo y retención cuya salida es la entrada del CAD. Este digitaliza la entrada analógica mediante aproximaciones sucesivas y el genera como resultado una palabra digital de 10 bits. Posee una entrada de voltaje de referencia en alto y otra en bajo las cuales resultan de alguna combinación de V_{DD} , V_{SS} , RA2 y RA3. (Microchip, 2003b).

El CAD utiliza en su configuración y funcionamiento 4 registros:

- ADRESH (byte más significativo de la palabra digital).
- ADRESL (byte menos significativo de la palabra digital).
- ADCON0 (registro de control de la operación del CAD).
- ADCON1 (registro de configuración de los pines del CAD).

Los registros de control y configuración se muestran a continuación:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

Figura 1.8. Registro de control de operación del CAD (ADCON0).

En PIC 16F877A el registro ADCON0 se encuentra en la posición 1F del banco 0 dentro del mapa de memoria y sus bits poseen las siguientes funciones:

- Bits 7-6: bits de selección del reloj de conversión.
 - 00: $F_{OSC}/2$
 - 01: $F_{OSC}/4$
 - 10: $F_{OSC}/8$
 - 11: F_{RC} (reloj derivado del oscilador RC interno del CAD).
- Bits 5-3: bits de selección del canal analógico.
 - 000: canal 0 (RA0, AN0).
 - 001: canal 1 (RA1, AN1).
 - 010: canal 2 (RA2, AN2).
 - 011: canal 3 (RA3, AN3).
 - 100: canal 4 (RA5, AN4).
 - 101: canal 5 (RE0, AN5).
 - 110: canal 6 (RE1, AN6).
 - 111: canal 7 (RE2, AN7).
- Bit 2: bit de estado de la conversión.
 - 1: conversión en progreso (al poner este bit a '1' se inicia la conversión).
 - 0: conversión finalizada (este bit se limpia por hardware cuando la conversión finaliza).
- Bit 1: no implementado.
- Bit 0: bit de habilitación del CAD.
 - 1: CAD habilitado.
 - 0: CAD deshabilitado (el CAD no tiene consumo de corriente).

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Figura 1.9. Registro de configuración del CAD (ADCON1).

Para el caso del PIC 16F877A el registro ADCON1 se encuentra en la posición 9F del banco 1 dentro del mapa de memoria y sus bits poseen las siguientes funciones:

- Bit 7 : bit de selección del formato de la palabra digital
 - 1: justificación a la derecha (los 6 bits mas significativos del registro ADRESH son leídos como '0').
 - 0: justificación a la izquierda (los 6 bits menos significativos del registro ADRESL son leídos como '0').
- Bit 6-4: no implementados.
- Bit 3-0: bits de configuración de los terminales de CAD donde A = entrada analógica y D = entrada/salida digital.

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

Los registros ADRESH: ADRESL contienen los 10 bits del resultado de la conversión. Cuando la conversión se completa el resultado se carga en este par de registros, el bit GO/DONE (ADCON0, 2) se pone a '0' y se activa la bandera de interrupción ADIF (PIR1, 6). El diagrama en bloques del CAD se muestra en la Figura 1.10.

Para realizar una conversión se deben seguir los pasos siguientes:

1. Configurar el CAD.
 - a. En el registro ADCON1 configurar las entradas y voltajes de referencia.
 - b. En el registro ADCON0 seleccionar el canal de entrada.
 - c. En el registro ADCON0 seleccionar el reloj del CAD.
 - d. En el registro ADCON0 habilitar el CAD.
2. Configurar la interrupción de fin de conversión (si se desea).
 - a. Limpiar el bit ADIF (PIR1, 6).
 - b. Poner a '1' el bit ADIE (PIE1, 6)
 - c. Poner a '1' el bit PEIE (INTCON, 6).
 - d. Poner a '1' el bit GIE (INTCON, 7).
3. Esperar el tiempo de adquisición del dato analógico de entrada.
4. Comenzar la conversión poniendo a '1' el bit GO/DONE (ADCON0, 2).
5. Esperar el fin de conversión mediante:
 - a. Encuesta del bit GO/DONE hasta que se limpie por hardware indicando el fin de conversión.
 - b. Esperar la interrupción de fin de conversión.
6. Leer el resultado del par ADRESH: ADRESL y limpiar la bandera ADIF si se requiere.
7. Para realizar una próxima conversión volver a paso 1 o 2 según se requiera.

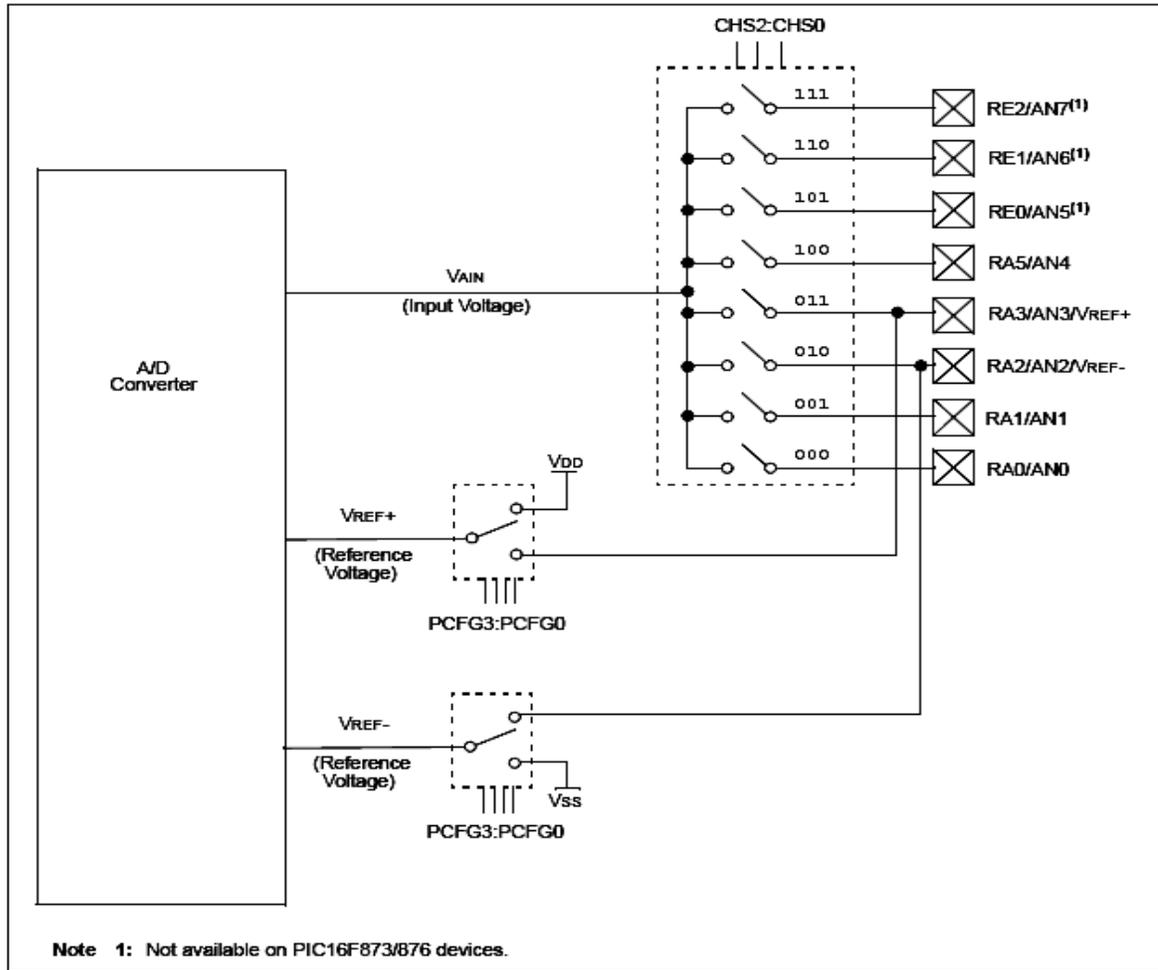


Figura1.10. Diagrama en bloques del CAD.

REQUERIMIENTOS DE CONVERSIÓN PARA EL CAD.

Para que el CAD consiga su exactitud especifica el capacitor del circuito de muestreo y retención (C_{HOLD}) se debe cargar al voltaje del circuito analógico de entrada, cuyo modelo se muestra en la Figura 1.11. Se define como tiempo de adquisición del dato el tiempo que permanece el capacitor C_{HOLD} conectado la fuente de voltaje externo. La impedancia de fuente externa (R_s) y la impedancia interna de muestreo (R_{SS}) afectan directamente el tiempo requerido para la carga de C_{HOLD} . La máxima impedancia para fuentes analógicas recomendada es de $2.5\text{k}\Omega$. Disminuyendo este valor se disminuye el tiempo de adquisición del dato. Una vez que se selecciona el canal analógico de entrada hay que tener en cuenta

el tiempo de adquisición del dato antes de iniciar la conversión como se muestra en la secuencia de conversión en la Figura 1.12.

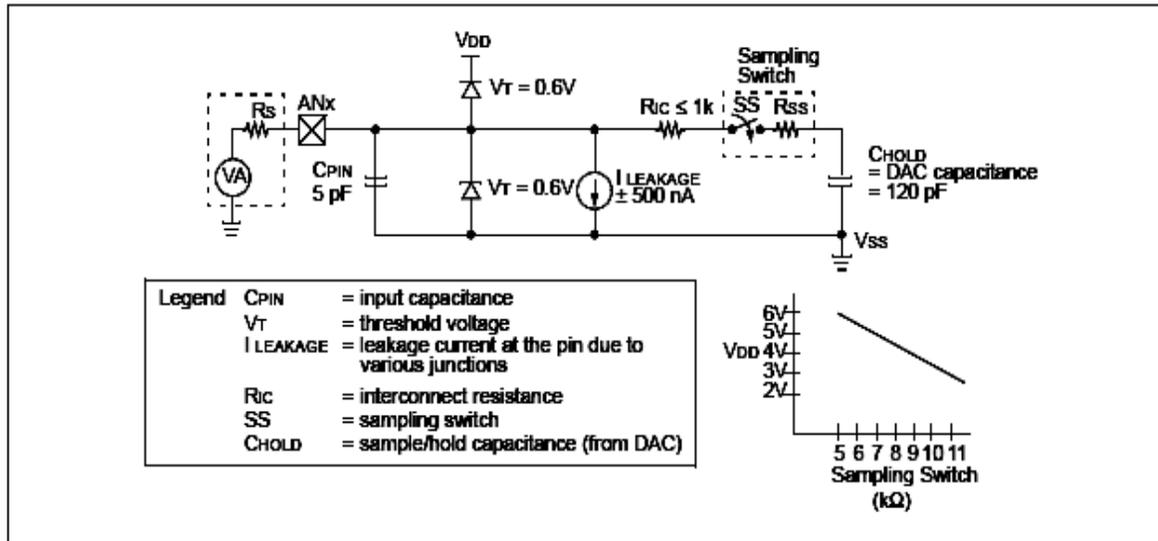


Figura 1.11. Circuito analógico de entrada del CAD.

Para calcular el mínimo tiempo de adquisición necesario se utiliza la ecuación 1.1.

T_{ACQ} = Amplifier Settling Time + Holding Capacitor Charging Time + Temperature Coefficient.

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF} \tag{1.1}$$

Esta ecuación asume un error de ½ LSB que es el máximo error permitido para lograr la resolución específica del CAD. El tiempo de carga del capacitor C_{HOLD} se calcula mediante la ecuación 1.2.

$$V_{HOLD} = (V_{REF} - (V_{REF} / 2048)) * (1 - e^{(-T / C_{HOLD} * (R_{IC} + R_{SS} + R_s))}).$$

O:

$$T_C = - (120pF) * (1 k\Omega + R_{SS} + R_s) * \ln (1/2047) \tag{1.2}$$

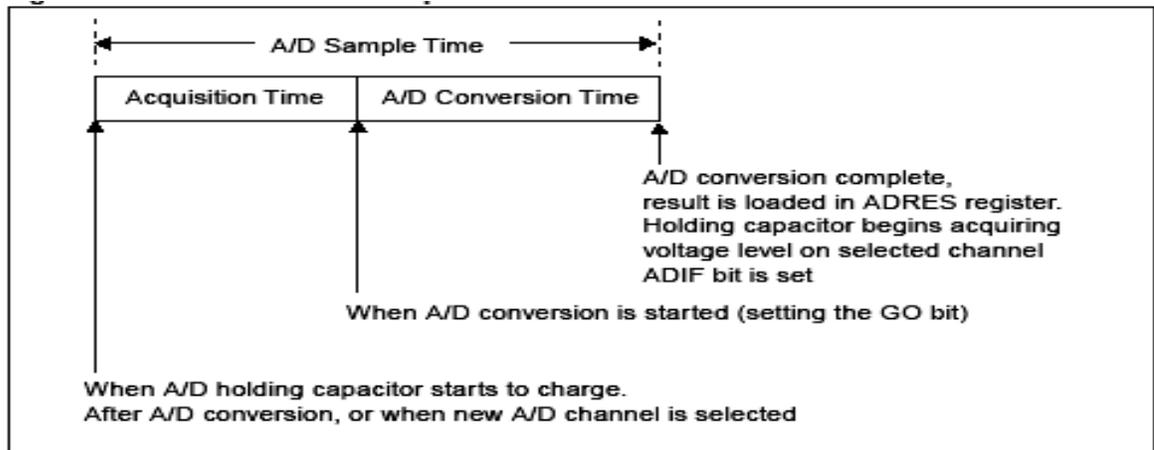


Figura 1.12. Secuencia de conversión para el CAD.

SELECCIÓN DEL RELOJ PARA EL CAD.

El tiempo de conversión por bit se define como T_{AD} . La conversión requiere un tiempo de $11.5 \cdot T_{AD}$ para 10 bits como se muestra en ciclo de conversión en la Figura 1.13. El reloj para la conversión se selecciona por software y las cuatro opciones posibles son:

- $2 \cdot T_{OSC}$.
- $8 \cdot T_{OSC}$.
- $32 \cdot T_{OSC}$.
- Reloj RC interno.

Para realizar una correcta conversión el reloj debe ser seleccionado de manera tal que asegure un mínimo tiempo T_{AD} de 1.6 μs . En la Tabla 1.3. se muestra la relación entre T_{AD} y la frecuencia de operación.

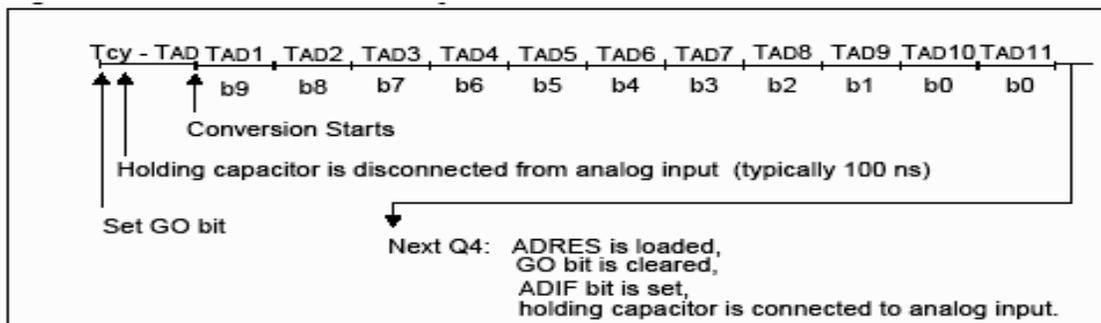


Figura 1.13. Ciclo de conversión para el CAD.

AD Clock Source (TAD)		Device Frequency			
Operation	ADCS1:ADCS0	4 MHz	2 MHz	1.25 MHz	333.33 kHz
2TOSC	00	500 ns ⁽²⁾	1.0 μ s ⁽²⁾	1.6 μ s ⁽²⁾	6 μ s
8TOSC	01	2.0 μ s ⁽²⁾	4.0 μ s	6.4 μ s	24 μ s ⁽³⁾
32TOSC	10	8.0 μ s	16.0 μ s	25.6 μ s ⁽³⁾	96 μ s ⁽³⁾
RC	11	3 - 9 μ s ^(1,4)			

Tabla 1.3. T_{AD} vs. Frecuencia de operación.

RESULTADO DE LA CONVERSIÓN ANÁLOGO-DIGITAL

El par ADRESH: ADRESL (16 bits) es la localización donde el resultado del CAD es cargado una vez finalizada la conversión. El CAD ofrece la flexibilidad de justificar el resultado de 10 bits en el par de 16 bits a la derecha o a la izquierda. El tipo de justificación es seleccionada por el bit ADFM (ADCON1, 7). La operación de justificación del resultado se muestra en la Figura 1.14. Los bits extra se ponen a '0'. Cuando el CAD no se utiliza el par ADRESH: ADRESL se puede utilizar como registros de propósito general.

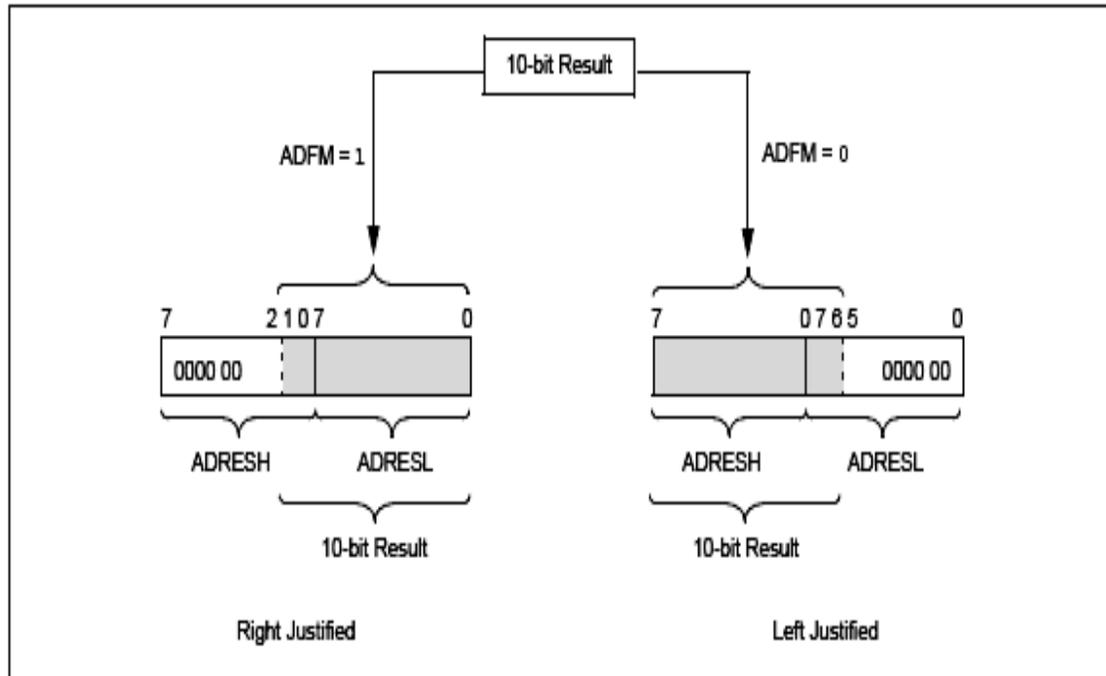


Figura 1.14. Justificación del resultado del CAD.

CAPÍTULO 2. Aplicaciones con subsistemas avanzados del PIC 16F877A.

En este capítulo se describen tres aplicaciones empleando el 16F877A. Dos de ellas utilizan los subsistemas avanzados descritos en el capítulo anterior. En cada aplicación se describe su esquema circuital (hardware) y el diagrama en bloques (software).

Para la confección y puesta a punto del software se utilizó la herramienta MPLAB de Microchip. (Microchip, 2003a)

2.1 Comunicación serie PC-PIC con visualización a través de display LCD.

En esta aplicación se emplea el interfaz serie(SCI) del PIC 16F877A descrita en el capítulo 1 el cual permite al mismo recibir mensajes enviados desde una computadora utilizando el Hyper Terminal el cual transmite los códigos ASCII de los caracteres que conforman el mensaje a través del puerto serie de la PC. La aplicación esta concebida de la siguiente manera: al energizar el sistema el display muestra un mensaje de espera: "Waiting serial communication" y a continuación el sistema se mantiene esperando que el usuario teclee un mensaje a través del Hyper Terminal. Cuando el usuario inicia la comunicación el mensaje de espera se continua mostrando hasta que el usuario indica el fin de mensaje con el caracter '.' (Punto). Los códigos ASCII recibidos se almacenan en memoria a partir de la dirección 40H y hasta la dirección 5FH (32 posiciones). Una vez que se recibe el código ASCII del punto, el sistema procede a borrar el mensaje de espera y a colocar el mensaje enviado por el usuario en el display. El sistema permanece en este estado hasta que un nuevo mensaje es enviado con lo cual se

repite el proceso. Cualquier reset o suspensión-restablecimiento de energía hará al sistema comenzar mostrando su mensaje de espera. (Valdés et al., 2007)

Esta aplicación en funcionamiento se muestra en el Anexo I.

2.1.1 Esquema circuital.

El hardware que permite implementar esta aplicación se muestra en la Figura 2.1.

Aunque la interfaz serie del PIC permite una comunicación bidireccional como se refleja en el esquema circuital en esta aplicación se emplea la misma en una sola dirección: de la computadora hacia el PIC. Para ello se hace uso del pin 2 del puerto serie de la computadora por el cual se transmiten los datos.

Debido a que el puerto serie trabaja con voltajes entre -12V (nivel bajo) y +12V (nivel alto) es necesario el uso del integrado MAX-232A el cual convierte estos niveles de voltaje a niveles lógicos: 0V (nivel bajo) y 5V (nivel alto), mediante la configuración mostrada en la figura 2.1. (Maxim Integrated Products, 2004; Palacherla, 1997)

Los datos son recibidos por el terminal 13 de la MAX-232A y se envían al PIC por el terminal 12. La interfaz serie del PIC recibe los datos por el terminal 26. El terminal 25 es que utiliza el PIC para transmitir y no se emplea en esta aplicación. Los códigos ASCII de los caracteres recibidos se almacenan en memoria a partir de la posición 40 H y hasta la posición 5F lo cual nos da un total de 32 caracteres que es la cantidad que permite visualizar el display LCD. Los terminales 15 (RC0), 16 (RC1) y 17 (RC2) están conectados con los terminales RS, RW y E del display LCD y se utilizan para controlar el funcionamiento del display y los 8 bits del puerto B (pines 33 al 40) se utilizan para enviar al display el código del carácter a visualizar por los terminales D0-D8. Como el display LCD utiliza para a visualización de los caracteres el código ASCII también no es necesario ningún tipo de procesamiento del dato recibido. Conforme se recibe se almacena en memoria y se envía al display para su visualización en el mismo formato. El resto de las conexiones tienen que ver con la alimentación de los elementos que conforman el hardware y están explícitas en el esquema circuital.

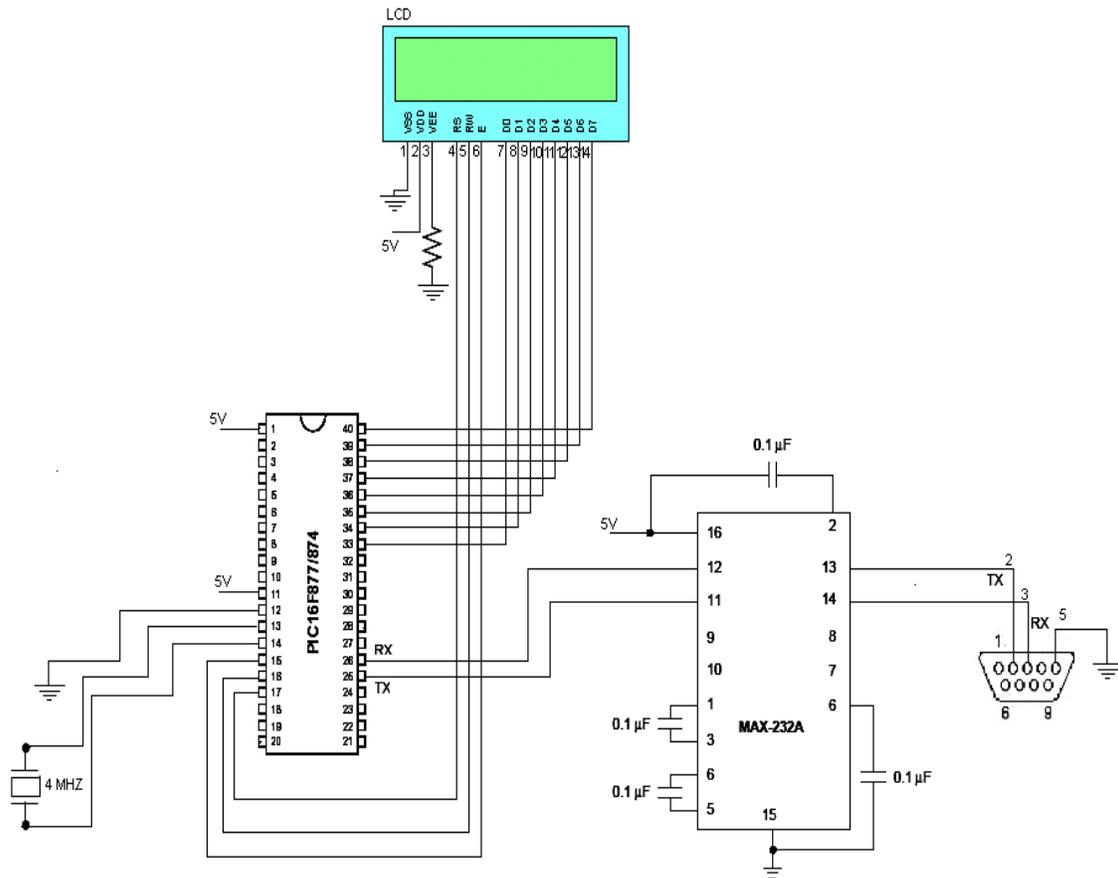


Figura 2.1. Comunicación serie PC-PIC con visualización a través de display LCD.

2.1.2 Diagrama en bloques y descripción del software.

El diagrama en bloques del software se muestra en la Figura 2.2 y a continuación se describe la función de cada subrutina utilizada en la aplicación.

- Subrutina INIREG: se configura el puerto B como salida que es usado para enviar los comandos de inicialización del display LCD y los datos a visualizar en el mismo. Se configuran los bits RC0, RC1 y RC2 como salidas, que se encuentran conectados a los terminales E, RW y RS del display LCD y permiten habilitar el display(E) y seleccionar el byte recibido por el puerto B como dato o comando(RS, RW). Se inicializa la variable

TablaPTR que constituye el puntero de la tabla de datos donde se almacenan los códigos ASCII de los caracteres recibidos.

- Subrutina INISER: esta subrutina habilita y configura el interfaz serie (SCI) configurando los bits RC6 como salida (terminal TX, no se emplea en la aplicación) y RC7 como entrada (terminal de RX). El SCI se configura con los siguientes parámetros: comunicación asincrónica, razón de Baud de 1200 Bps y habilitación de la interrupción de recepción.
- Subrutina INIDIS: esta subrutina inicializa el display y lo deja habilitado para mostrar mensajes. Para ello se sigue una secuencia establecida por el fabricante en la hoja de datos del display LCD. (Segredo, 2006).
- Subrutina MESSAGE: muestra en el display el mensaje de espera.
- Subrutina de interrupción: lee el dato recibido por el SCI del registro RCREG y lo almacena en la variable DATO.
- Subrutina Llenar: esta subrutina va almacenando en memoria los códigos ASCII recibidos por el SCI. Este proceso se realiza utilizando direccionamiento indirecto: el valor del puntero TablaPTR se carga en el registro FSR y se escribe en cada posición apuntada con el registro INDF. Una vez recibido el punto se culmina el almacenamiento y se reinicializa el puntero TablaPTR.
- Subrutina Limpiar: elimina el mensaje del display LCD escribiendo el caracter espacio en las 32 posiciones del display.
- Subrutina MESSAGE1: muestra el contenido del mensaje almacenado en la memoria.

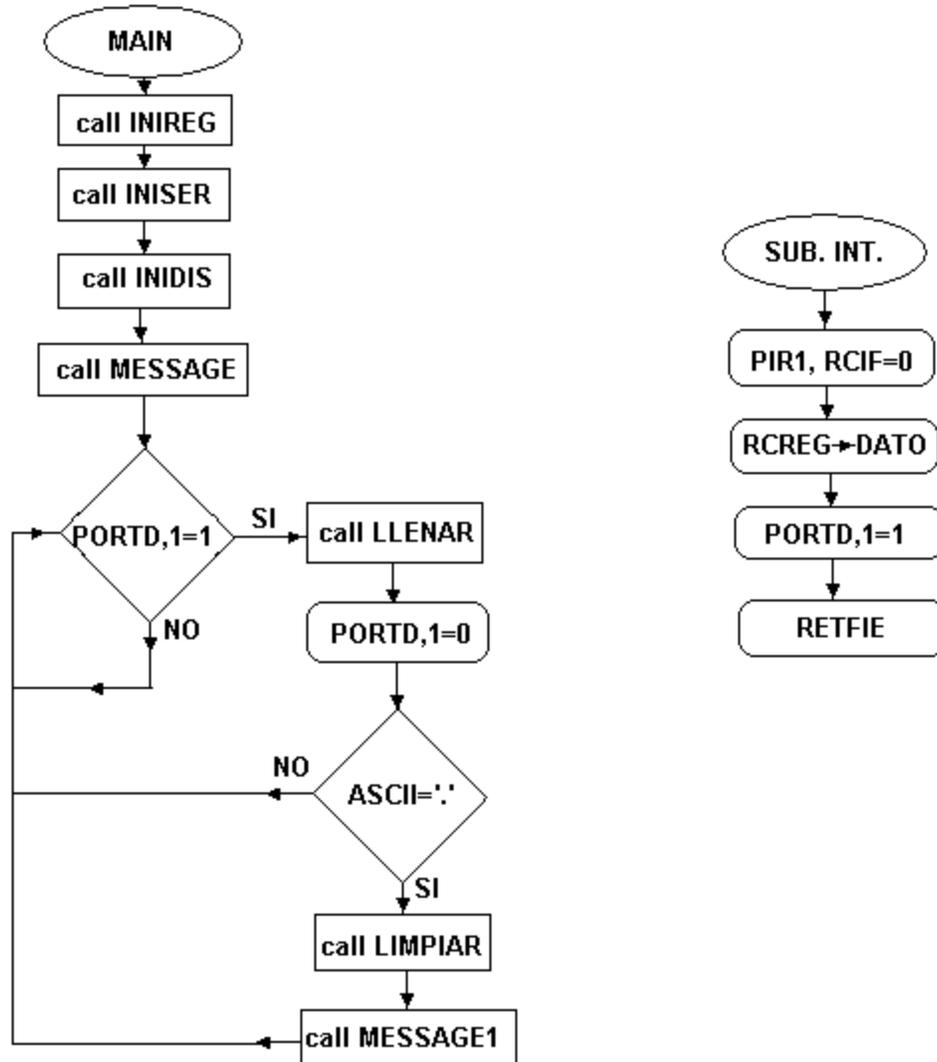


Figura 2.2. Diagrama en bloques de la comunicación serie PIC-PC.

2.2 Medición de temperatura con visualización a través de display LCD.

Esta aplicación consiste en visualizar en el display la temperatura ambiente. Para ello se utiliza un sensor LM35 el cual entrega a su salida un voltaje proporcional a la temperatura sensada con la proporción de $10\text{mV}/^{\circ}\text{C}$ (por ejemplo para una temperatura de 25°C el sensor entrega a la salida 250mV). (National Semiconductor, 1997)

La interfaz de conversión análogo-digital (CAD) del PIC 16F877A descrita en el capítulo 1 se encarga de digitalizar este valor y subsiguientes subrutinas lo procesan, validan y finalmente envían el resultado al display LCD para su visualización. En cada fin de conversión se actualiza el valor de temperatura en el display. En caso que el resultado no se encuentre entre los límites de validación el display se mantiene con el resultado anterior. (Tebus et al., 2007)

La aplicación en funcionamiento se muestra en el Anexo II.

2.2.1 Esquema circuitual

El hardware de esta aplicación se muestra en la figura 2.3 y se describe a continuación.

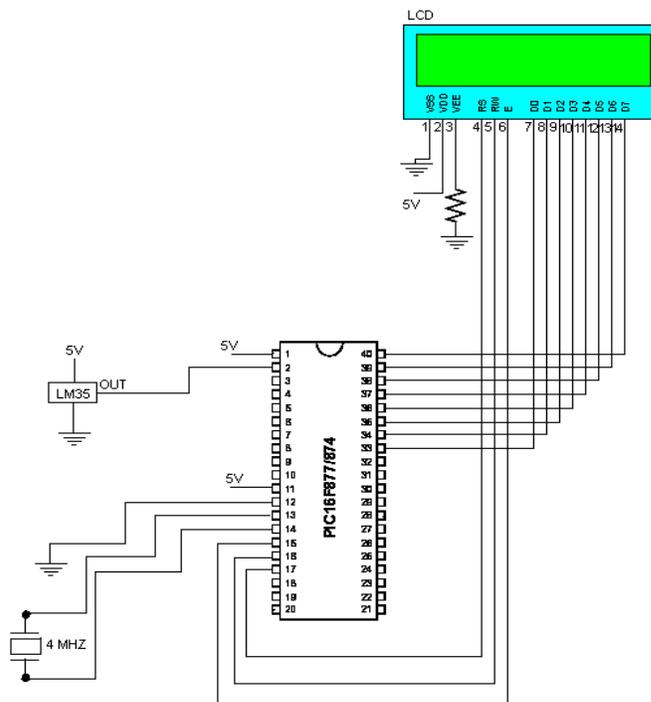


Figura 2.3. Medición de temperatura con visualización a través de display LCD.

El sensor LM35 posee tres terminales: Vcc, tierra y salida(out); la cual se encuentra conectada al terminal 2 del PIC que es la entrada al canal 0 del CAD

(AN0) el cual es seleccionado por software. El puerto B (pines 33-40) se conecta a los terminales D0-D7 del display LCD y se utiliza para enviar los comandos de inicialización o los datos que se van a visualizar. El terminal de habilitación (E) y los que se encargan de seleccionar el byte D0-D7 como comando o datos (RS, RW) se encuentran conectados a los terminales RC0, RC1 y RC2 (pines 15-17), de manera similar a las aplicaciones descritas en los epígrafes anteriores. El resto de las conexiones consisten en la alimentación de los elementos que conforman el circuito y están explícitas en el mismo.

2.2.2 Diagrama en bloques y descripción del software.

Para la elaboración del software que resuelve esta aplicación se partió de las características del CAD descritas en el capítulo anterior para obtener el diagrama en bloques del sistema que se muestra en la figura 2.4 y se describe a continuación especificando la función de cada subrutina empleada.

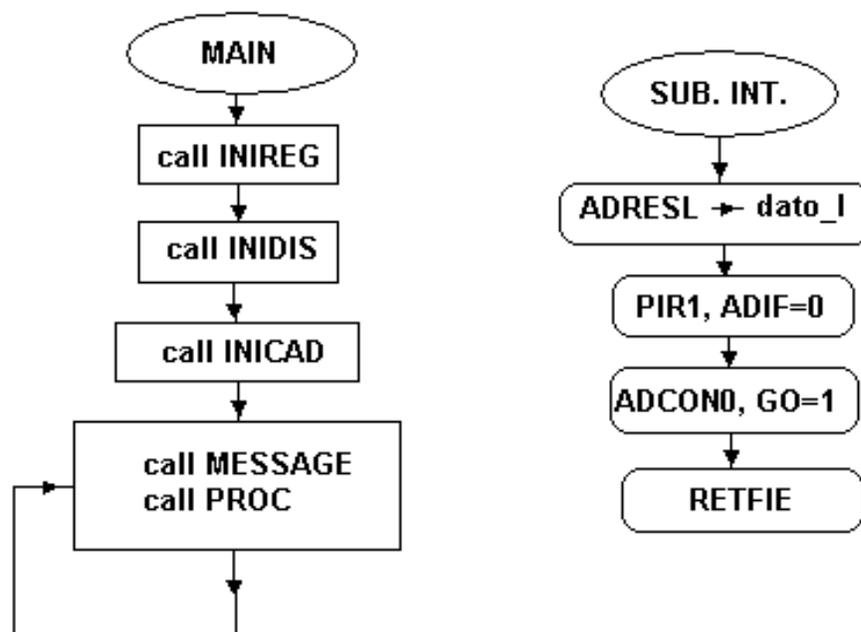


Figura 2.4. Diagrama en bloques de la medición de temperatura.

Subrutinas empleadas en la aplicación:

- Subrutina INIREG: En esta subrutina se inicializan los registros con la siguiente configuración: puerto B y puerto D como salidas todos sus terminales que se encuentran conectados al display LCD. Se inicializan además los registros que se utilizan en la subrutina de procesamiento de la señal.
- Subrutina INIDIS: esta subrutina inicializa el display y lo deja habilitado para mostrar mensajes. Para ello se sigue una secuencia establecida por el fabricante en la hoja de datos del display LCD. (Segredo, 2006).
- Subrutina INICAD: en esta subrutina se habilita y se configura el CAD con las siguientes características: dato entregado por el CAD con justificación a la derecha, o sea los 8 bits del registro ADRESL y los dos bits menos significativos del registro ADRESH, reloj RC para el CAD y selección del canal analógico AN0. Se habilita la interrupción por fin de conversión de CAD y la interrupción global y de periféricos. Finalmente se da la orden de inicio de conversión. A continuación el sistema se mantiene en un lazo de visualización hasta que se culmina la conversión y se activa la interrupción del CAD.(Microchip, 1997)
- Subrutina Message: en esta subrutina se adquiere el dato, se procesa, se valida y se muestra en el display LCD con el siguiente formato: XX°C. Del procesamiento del dato se encarga a la subrutina PROC la cual se llama desde la subrutina Message.
- Subrutina PROC: en esta subrutina se adquiere el dato entregado por el CAD cuando culmina la conversión, se procesa, se valida y se entrega a la subrutina Message para su visualización. El CAD entrega un dato de 10 bits y como se escogió justificación a la derecha dichos bits se encuentran en el registro ADRESL (8 menos significativos) y en ADRESH (2 as significativos). Como la medición de temperatura se realiza en niveles de temperatura ambiente, los valores de temperatura se encuentran en un

rango de 5-40 °C, teniéndose en cuenta las características del clima en nuestro país. A partir de la ecuación de conversión :

$$V_D = [(2^n - 1)/V_{max}] * V_{in} \quad (2.1)$$

Con los siguientes valores para nuestra aplicación particular:

$$n = 10 \quad (\text{CAD de 10bits}).$$

$$V_{max} = 5V \quad (\text{el } V_{cc} \text{ del sistema es de 5V})$$

$$V_{in} = 0.01 * T \quad (\text{el sensor LM35 entrega 10mV/°C})$$

Y calculando para un valor extremo de temperatura de 40°C tenemos que:

$$V_D = [(2^{10} - 1)/5] * 0.01 * 40$$

$$V_D = 81.84$$

O sea que para nuestra aplicación particular el voltaje digital entregado por el CAD podrá ser representable por 8bits (reacuérdesse que con 8bits se pueden representar valores entre 0 y $2^8 - 1$ o sea entre 0 y 255), por tanto y como se usa justificación a la derecha los bits mas significativos que se encuentran en ADRESH se podrán despreciar y el valor de V_D se puede leer de los 8 bits menos significativos que se encuentran en el registro ADRESL.

El procesamiento de la señal consiste en convertir este valor de voltaje digital en el valor de temperatura que el representa. Aplicando 2.1 a nuestro caso particular tenemos:

$$V_D = [(2^{10} - 1)/5] * 0.01 * T$$

Lo cual nos lleva a la siguiente ecuación de conversión:

$$V_D = 2.046 * T \quad (2.2)$$

La cual es consistente con el caracter lineal de la característica de salida del LM35. Invertiendo la abcisa y la ordenada de la ecuación anterior tenemos que:

$$T = (1/2.046)*V_D$$

Resolviendo:

$$T = 0.488756 * V_D \quad (2.3)$$

Que es la ecuación que se utiliza para la conversión del voltaje digital en el valor de temperatura a visualizar en el display LCD. Ahora bien para simplificar un poco las operaciones matemáticas y teniendo en cuenta que el valor de temperatura a mostrar será solo un entero podemos redondear la ecuación 2.3 sin que esto introduzca un error apreciable en el resultado. Esto se puede demostrar calculando el error para el valor extremo superior de 40°C:

$$\text{Error } 0.5 * 40^\circ\text{C} - 0.488756 * 40^\circ\text{C}$$

$$\text{Error } 0.449^\circ\text{C}$$

Como se puede apreciar aún para el valor extremo superior de temperatura el error introducido es menor que 0.5 °C, esto unido a que el valor de temperatura va a ser mostrado en números enteros hace que este error se pueda despreciar.

De ahí que:

$$T = 0.5 * V_D$$

Escrito de otra manera:

$$T = V_D / 2 \quad (2.4)$$

Como estamos trabajando en base hexadecimal (binario) la operación de división entre potencias de 2 consiste en rotar a la derecha el valor de V_D tantas veces como potencia de 2 sea el divisor. Como en este caso el divisor es 2 el valor de V_D se rota una vez hacia la derecha, perdiéndose de esta manera el bit menos significativo del valor de V_D lo cual es el resultado de la operación de redondeo de la ecuación (2.3). De esta forma el

procesamiento de la señal se simplifica y el software adquiere menor complejidad, consistiendo el mismo en:

- Rotar V_D a la derecha una vez.
- Dividir el resultado obtenido entre 10 para separar unidades y decenas para visualizar valor de temperatura. Al dividir entre 10 el cociente corresponde a las decenas y el resto a las unidades.
- Validar la temperatura con los siguientes límites: decenas entre 0 y 5; unidades entre 0 y 9.

Finalmente obtener el código ASCII de los valores de las decenas y unidades y enviarlo al display LCD para su visualización.

2.3 Comunicación teclado-PIC con visualización a través del display LCD.

Esta aplicación implementa la comunicación de un teclado con el PIC 16F877A lo cual permite recibir mensajes y visualizarlos en un display LCD. Con una filosofía similar a la comunicación serie PC-PIC el PIC recepciona los códigos ASCII de los caracteres del mensaje tecleado por el usuario y los va almacenando en memoria a partir de la dirección 40H y hasta la dirección 5FH (32 posiciones). Al recibir energía el sistema muestra el mensaje de espera "Waiting keyboard message" hasta que el código ASCII del carácter '.' (punto) es recibido, indicando el fin del mensaje tecleado por el usuario. A continuación se limpia el display y se muestra dicho mensaje de usuario, el cual se mantiene hasta que otro mensaje es tecleado. Cualquier reset o suspensión-restablecimiento de energía hará al sistema comenzar mostrando su mensaje de espera. (Morell et al., 2007)

La aplicación en funcionamiento se muestra en el Anexo III.

2.3.1 Esquema circuital

Para concebir el hardware para esta aplicación se tuvo que analizar la estructura de los datos enviados por el teclado que se muestra en la figura 2.5. Cada dato consiste en dos señales: una señal de reloj (terminal 5 del conector PS/2-hembra) que en estado de reposo se encuentra en alto (5V) y una señal de datos (terminal 3 del conector PS2-hembra) que igualmente en estado de reposo se encuentra en alto (5V). Cuando el teclado va a enviar un dato coloca una onda cuadrada en el terminal de reloj y en cada flanco de bajada de dicha señal de reloj coloca un bit en la señal de datos con la siguiente secuencia: bit de inicio-8 bits de datos-bit de paridad-bit de parada. (Chapweske, 2003)

Con el objetivo de poder desarrollar una aplicación óptima se decidió conectar el terminal de reloj al terminal de interrupción externa (RB0) de manera tal que se detecte una interrupción en cada flanco de bajada de la señal de reloj. Cada vez que ocurre una interrupción se lee el bit correspondiente en la señal de datos que se encuentra conectada en el terminal RB7 y se realiza un procesamiento de la señal descrito en el software del sistema.

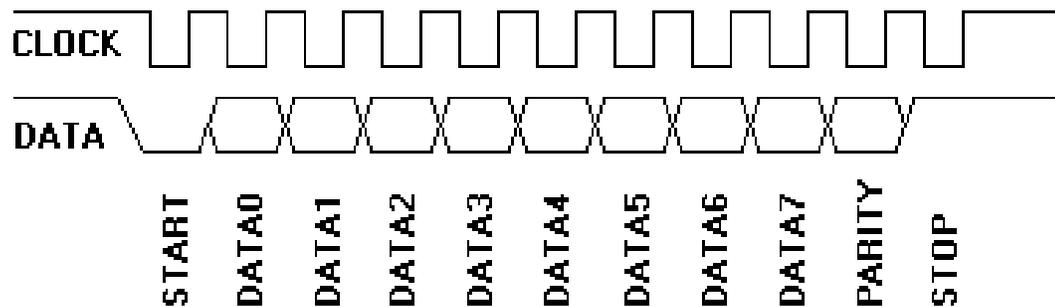


Figura 2.5. Estructura del dato enviado por el teclado.

Partiendo de los análisis anteriores el esquema circuital se muestra en la figura 2.6.

A diferencia del hardware de la comunicación serie PC-PIC para esta aplicación se utiliza el puerto B para recepcionar el código proveniente del teclado, colocándose la señal de reloj en el terminal RB0 (pin 33) la señal de datos en el

terminal RB7 (pin 40). Por consiguiente se utilizó el puerto D (pines 19-22 y 27-30) para el envío de las señales de control y datos a los terminales D0-D7 del display LCD. Se mantiene la conexión de los terminales RC0, RC1 y RC2 (pines 15-17) con los terminales E, RS y RW del display con los cuales se habilita el mismo y se selecciona la palabra en D0-D7 como dato o comando. El resto de las conexiones tienen que ver con la alimentación de los elementos del circuito y están explícitas en el diagrama circuital.

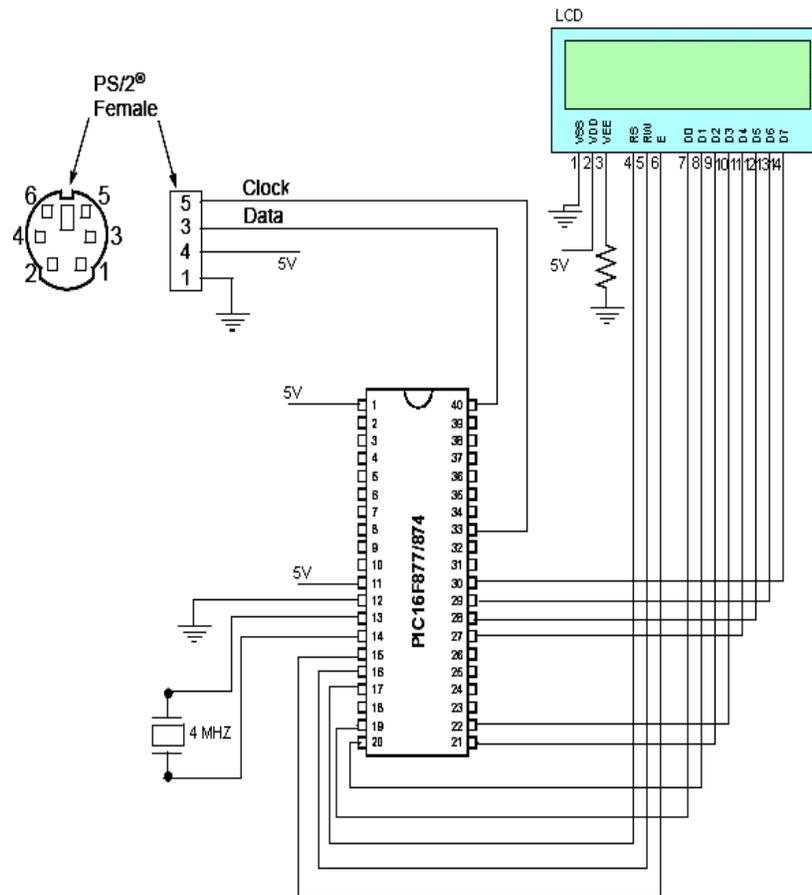


Figura 2.6. Comunicación teclado-PIC con visualización a través de display LCD.

2.3.2 Diagrama en bloques y descripción del software

Debido a su similitud con la comunicación serie PIC-PC en cuanto a filosofía el diagrama en bloques de esta aplicación resulta muy similar teniendo su diferencia mas marcada en el algoritmo de recogida del dato proveniente del teclado el cual se implementa en la subrutina de atención a la interrupción. El diagrama en bloques se muestra en la figura 2.7.

Las subrutinas empleadas en el software se describen a continuación:

- Subrutina INITEC: se configuran los puertos de la siguiente manera: puerto D y terminales RC0, RC1 y RC2 como salidas debido a que se encuentran conectados a los terminales D0-D7 y E, RS y RW del display LCD respectivamente; el puerto B se configura con sus terminales RB0 y RB7 como entradas, las cuales van a recepcionar las señales de reloj y datos respectivamente provenientes del teclado y el resto como salidas (no se utilizan en esta aplicación). Se habilita la interrupción externa RB0 y se inicializa el puntero a la tabla de datos tabla PTR en 40H.
- Subrutina INIDIS: esta subrutina inicializa el display y lo deja habilitado para mostrar mensajes. Para ello se sigue una secuencia establecida por el fabricante en la hoja de datos del display LCD. (Segredo, 2006).
- Subrutina MESSAGE: muestra en el display el mensaje de espera.
- Subrutina de interrupción: en esta subrutina es donde radican las principales diferencias de esta aplicación con respecto a la comunicación serie PC-PIC. Mientras que el SCI del PIC 16F877A almacena el dato recibido en el registro RCREG, siendo solo necesario leerlo una vez que ocurre la interrupción, en este caso para almacenar el dato se necesita de un algoritmo mas complejo.

Primeramente debemos considerar el hecho de que el dato enviado por el teclado posee 11bits por lo tanto cada dato produce 11 interrupciones. En cada interrupción se recoge el bit de RB7. El bit de inicio se desprecia. Los bits de datos se almacenan en un registro siempre en su posición más significativa y se van rotando a la derecha en una cantidad descendente

comenzando por 7. Al final de cada una de estas 8 interrupciones se salva el resultado de las operaciones realizadas en la variable BYTE. Finalmente los bits de paridad y parada se desprecian igualmente, quedando de esta manera el código enviado por el teclado en BYTE. Ahora este código recepcionado no es precisamente el código ASCII del caracter oprimido sino un SCANCODE, siendo necesario una subrutina que convierta dicho SCANCODE en código ASCII según la equivalencia mostrada en las tablas de los Anexos IV y V. Dicho código ASCII obtenido se almacena en memoria para su posterior envío al display LCD.

- Subrutina Llenar: esta subrutina va almacenando en memoria los códigos ASCII obtenidos. Este proceso se realiza utilizando direccionamiento indirecto: el valor del puntero TablaPTR se carga en el registro FSR y se escribe en cada posición apuntada con el registro INDF. Una vez recibido el punto se culmina el almacenamiento y se re-inicializa el puntero TablaPTR.
- Subrutina Limpiar: elimina el mensaje del display LCD escribiendo el caracter espacio en las 32 posiciones del display.
- Subrutina MESSAGE1: muestra el contenido del mensaje almacenado en la memoria.

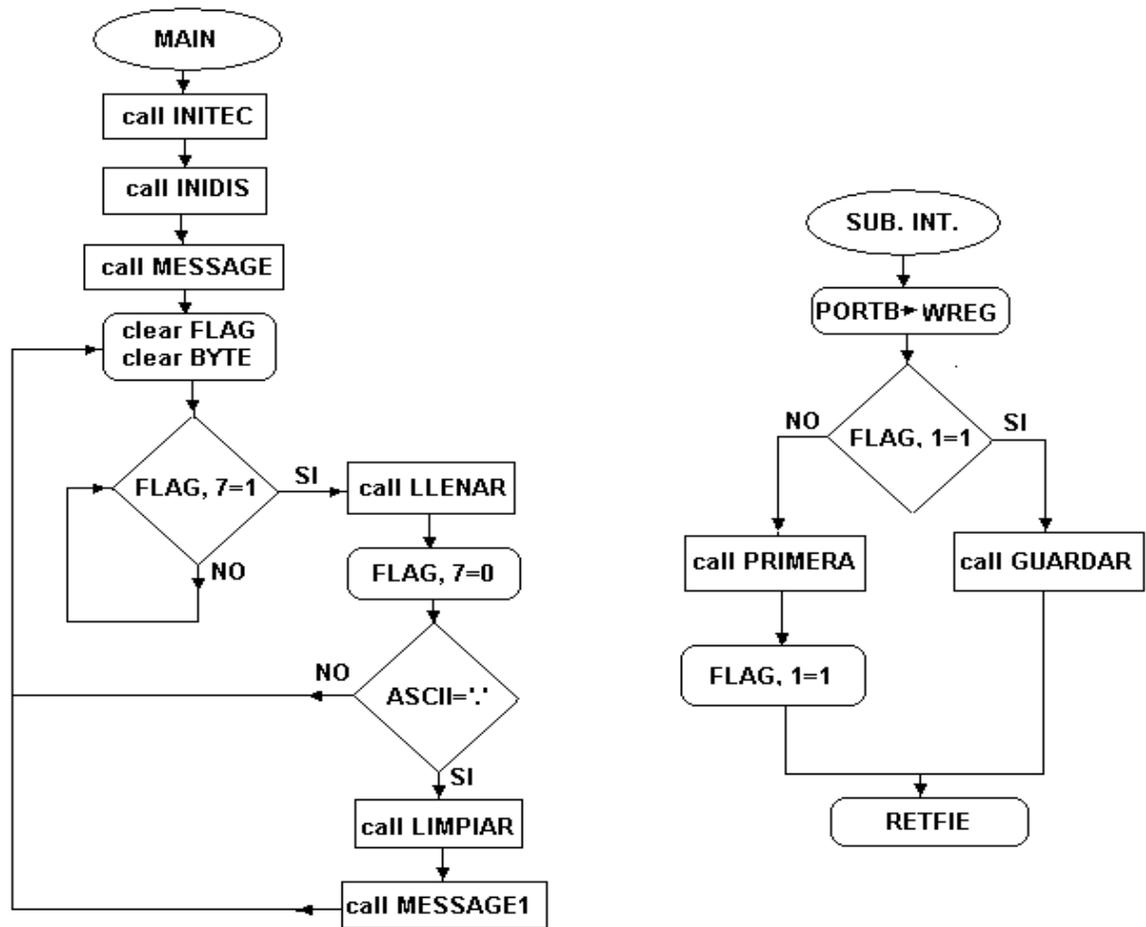


Figura 2.7. Diagrama en bloques de la comunicación teclado-PIC.

CAPÍTULO 3. Diseño, programación y montaje del cartel pasa-mensajes.

Se aborda el diseño y construcción del cartel pasa-mensajes. Para ello se realiza un análisis de las características de estos carteles y sus principales funciones, descripción detallada de la selección y diseño del hardware. Finalmente el diseño e implementación del software y el análisis económico.

3.1 Generalidades.

Los carteles pasa-mensajes electrónicos son un medio impactante y práctico para comunicar mensajes de información. Entre sus múltiples usos podemos mencionar:

1. Señalizar con dinamismo y practicidad.
2. Comunicar información en tiempo real.
3. Captar la atención.
4. Hacer todo esto y al mismo tiempo decorar nuestro entorno.(Indicart Carteles Electrónicos, 2001)

Existen diversos modelos en el mercado ya que son un medio de comunicación muy accesible y con gran alcance. Son una de las formas electrónicas más comunes para publicitar en cualquier tipo de comercio. Dentro de sus aplicaciones se destacan:

- Carteles para colegios y negocios que están buscando formas económicas de comunicarse con la comunidad.

- Carteles para Bancos que quieren informar a sus clientes dentro y fuera de sus edificios.
- Carteles para aeropuertos, estaciones de autobuses y departamentos de transportación que necesitan un sistema informativo con pantallas para ayudar a viajeros a llegar a sus destinos fácil y seguramente, a tiempo.
- Carteles para eventos deportivos internacionales, que requieren sistemas de anotación sofisticados para distintas competencias en sitios diferentes.

Para todas las aplicaciones antes mencionadas podemos encontrar carteles de todas las formas y tamaños que poseen una diversa variedad de colores en sus presentaciones, ya sean simple línea, doble o multilínea. En dependencia del tipo de aplicación para la que se use el cartel y las facilidades que este brinde será su costo, los cuales van desde los 200.00 USD hasta los 2000.00 USD y más. (Mercado Libre, 2007)

3.2 Diseño y análisis del Hardware.

3.2.1 Características generales.

El cartel esta formado por un display de 8 matrices de 8X8 diodos LEDs una a continuación de la otra, que permite visualizar símbolos, ya sea caracteres que forman palabras o números. Posee una programación que permite mostrar mensajes en movimiento que se mueven de derecha a izquierda. Las columnas de las matrices están atendidas por barrido, a través de transistores que son activados por registros de desplazamiento. Las 8 filas del display se encuentran conectadas en paralelo y son atendidas por un puerto del Microcontrolador PIC 16F877A como se muestra en la Figura 3.1. (Pino, 2004)

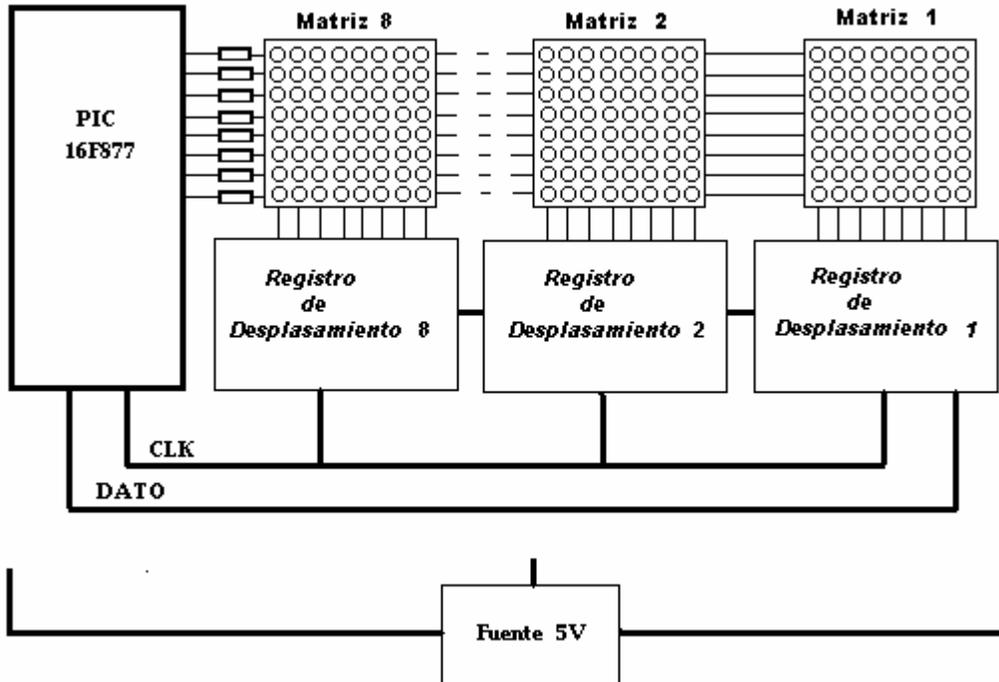


Figura 3.1. Diagrama en bloques general del cartel pasa-mensajes.

3.2.2 Matrices de LEDs.

Las matrices de diodos LEDs que se utilizan para el diseño del hardware tienen 8X8 LEDs conectados matricialmente. Poseen un total de 64 LEDs por cada matriz, 512 LEDs en el display en total. La conexión eléctrica de los LEDs está dispuesta de tal forma que las columnas tienen cátodo común y las filas ánodo común como se muestra en la Figura 3.2. (López, 2005)

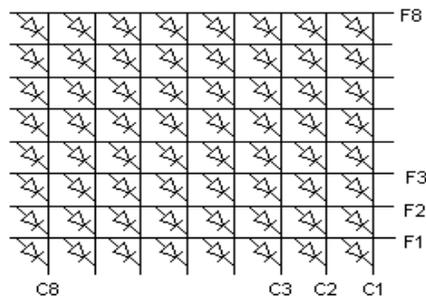


Figura 3.2. Conexión eléctrica de la matriz de LEDs.

Cada diodo LED posee una corriente para su adecuado encendido y su prolongada vida útil de 20mA con una caída de tensión de 1.26V.

Hay que tener en cuenta a la hora del diseño del impreso que la disposición de los terminales de las matrices no se encuentra de manera ordenada. La Figura 3.3 muestra el diagrama que identifica la posición de cada terminal.

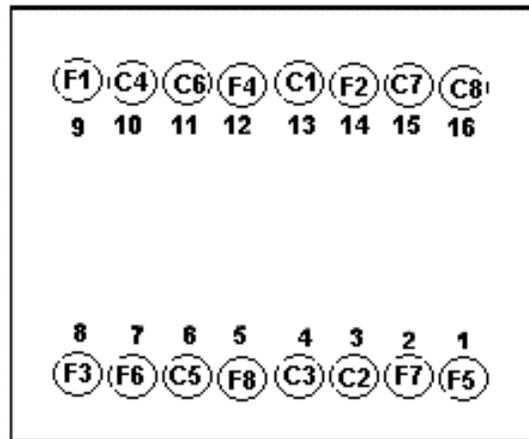


Figura 3.3. Identificación de terminales en la matriz de LEDs (vista posterior).

3.2.3 Registros de desplazamiento.

Como elemento fundamental para la realización del barrido se escogió el registro de desplazamiento 74LS164N, cuyo funcionamiento describimos a continuación.

El 74LS164N es uno de los circuitos integrados mas utilizados en proyectos de electrónica digital. Es un versátil registro de desplazamiento de 8 bits, también conocido como conversor serie-paralelo. Físicamente, se trata de un circuito integrado de 14 pines, en formato DIP, compuesto por 8 salidas. (Fairchild Semiconductor, 2000)

El diagrama de pines del registro de desplazamiento se muestra en la Figura 3.4.

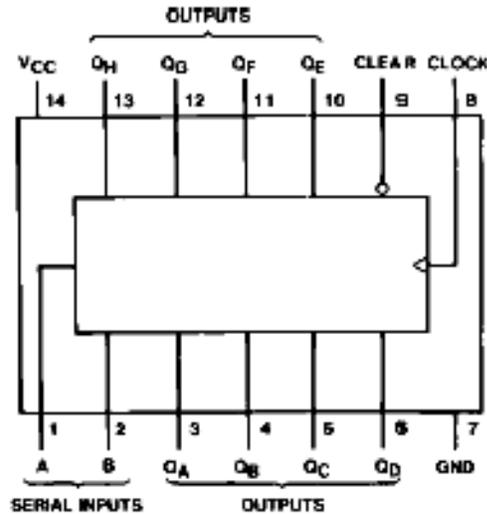


Figura 3.4. Diagrama de pines del registro de desplazamiento 74LS164.

Internamente el registro de desplazamiento 74LS164N contiene 8 flip-flops en cascada como se muestra en la Figura 3.5. Los bits ingresan al primero de ellos en cada pulso de reloj, y se propagan hacia la derecha. Si se comienza con todos los flip-flops en cero, lo cual se logra poniendo a '0' el pin reset durante unos nanosegundos, luego de 8 ciclos de reloj, tendremos disponibles en las salidas QA a QH los 8 bits ingresados por el terminal de datos. (terminales A y B cortocircuitados).

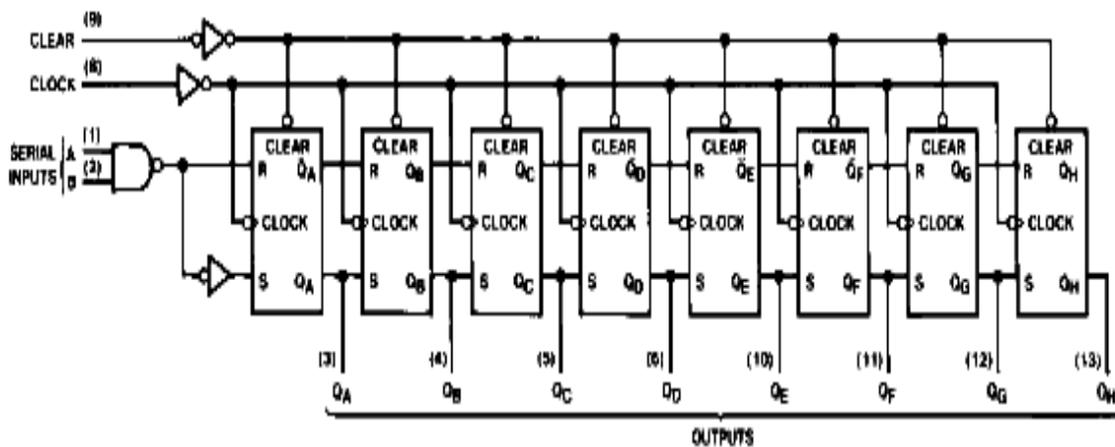


Figura 3.5 Diagrama lógico del 74LS164.

En el noveno pulso de reloj, se ingresara un nuevo dato al registro, y el que estaba presente en el ultimo flip-flop (salida QH) se perderá.

A menudo se necesita construir un registro de desplazamiento de más de 8 bits de profundidad como es el caso, se puede conectar la salida QH de un 74LS164N a las entradas A y B del segundo, de manera que este noveno bit que se hubiera perdido al “caerse” del primer 74LS164N se aproveche como dato de entrada en el siguiente. De esta manera, encadenando varios de estos integrados se puede construir un registro de desplazamiento del largo que se desee. Por supuesto, en el cartel pasa-mensajes las entradas de reloj de los 8 registros implicados deben estar unidas, para que funcionen sincronizadamente.

Los datos se introducen en el registro de desplazamiento con cada flanco de subida del pulso de reloj, esto es, cuando el pulso de reloj pasa de ‘0’ a ‘1’. Por ultimo, se debe tener en cuenta que el primer bit ingresado es el que primero saldrá por el otro extremo del registro, por lo que se dice que es una cola de bits del tipo FIFO (primero en entrar, primero en salir).

La corriente de salida en bajo (I_{OL}) que nos da el registro de desplazamiento no es suficiente para manejar cada columna durante el barrido. Para solucionar tal problema se conectó un transistor que fuera capaz de manejar la corriente de conducción de 8 LEDs al mismo tiempo. Para ello se escogió el BC546. Se tomó el puerto D del microcontrolador PIC16F877A que puede entregar una corriente de 25mA por terminal para las palabras binarias que activan las filas.

Para el cálculo de los resistores que limitan la corriente por el puerto se procede fijando una corriente máxima de 20mA, conociendo la caída de tensión en los LEDs de 1.26V y la caída colector-emisor de 0.2V se plantea una LKV que quedaría de la siguiente manera:

$$\mathbf{-5V + I_{RD7} + 1.26V + 0.2 = 0} \quad \mathbf{(3.1)}$$

Despejando,

$$RD7 = \frac{5V - 1.26V - 0.2V}{20mA} = \frac{3.54V}{20mA} = 177\Omega$$

Se escogió una resistencia de 180Ω que resulta ser el valor comercial más cercano, de estos resistores se necesitan 8, uno para cada terminal del puerto.

Utilizamos resistencias de $3.9K\Omega$ para garantizar que la salida en alto del registro de desplazamiento sea capaz de saturar los transistores encargados de manejar la corriente por columna. Se tienen que utilizar 64 resistores de este tipo. La Figura 3.6 facilita la comprensión del cálculo efectuado.

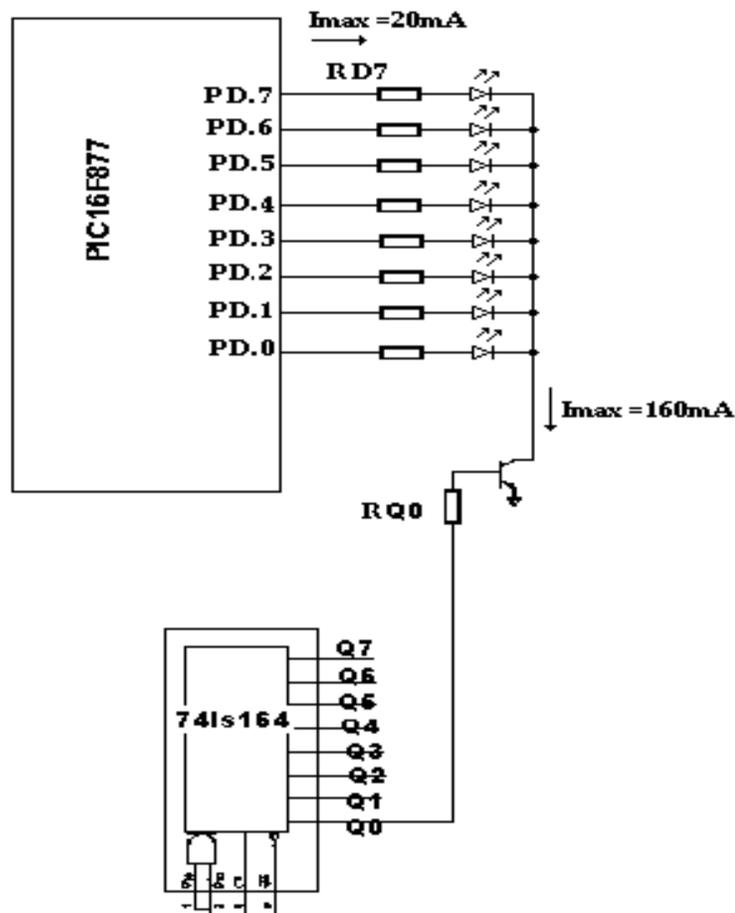
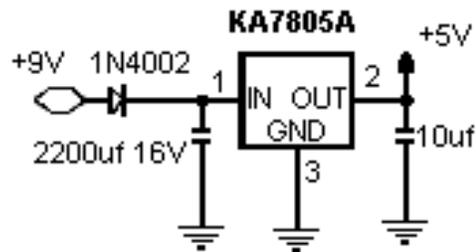


Figura 3.6. Conexión del registro de desplazamiento y el transistor BC546 con una columna de la matriz de LEDs y el PIC16F877.

3.2.4 Fuente de alimentación.

Para el cartel se utiliza una fuente regulada de 5V, tensión con la que trabajan los registros de desplazamiento, MAX232 y el PIC16F877. Por el bajo consumo del cartel se diseña una fuente regulada con KA7805A que regula el voltaje de un transformador exterior de 9V de corriente directa. (Fairchild Semiconductor, 2003)

En la Figura 3.7 se muestra el esquema eléctrico de la fuente de alimentación.



3.7. Esquema eléctrico de la fuente de alimentación.

3.2.5 Montaje y pruebas.

Una vez diseñado el circuito general del cartel pasa-mensajes que se muestra en la Figura 3.8, se realizaron diferentes pruebas que nos llevaron a construir con toda seguridad el circuito impreso. Se tomaron como base los esquemas de hardware de las aplicaciones descritas en el capítulo 2.

Las principales pruebas realizadas fueron:

1. Se montó en una placa de prueba las matrices con los registros de desplazamiento y se le hicieron pruebas de barrido con todas las filas activadas hasta lograr el encendido del display por completo. Esto dio una idea del valor de la frecuencia de barrido que se necesitaba y de la manera en que se trabaja con el ciclo útil de la señal de reloj para lograr una mayor intensidad en el encendido.

La implementación del software se realiza a bajo nivel con el lenguaje ensamblador y la herramienta MPLAB de Microchip que permite la compilación, depuración y puesta a punto del programa. (Microchip, 2003a)

3.3.1 Diagrama de bloques de las principales subrutinas y del programa principal.

Las principales subrutinas que conforman el software del cartel pasa-mensajes se describen a continuación:

- Subrutina BUFFER: es una de las subrutinas más sencillas del programa. La misma recibe en el registro DATO la palabra binaria correspondiente a la columna del símbolo que se va a visualizar, ya sea un carácter o un número y la coloca con direccionamiento indirecto en la posición 0EDH. Llama a la subrutina DISPLAY para que la visualice y luego a la subrutina ROTAR para que la desplace a la anterior posición de memoria. En esta subrutina se encuesta la bandera PIR1, 0 para la actualización del reloj en tiempo real. A continuación en la Figura 3.9 se muestra el diagrama en bloques de esta subrutina.

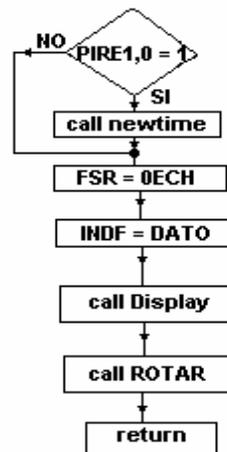


Figura 3.9. Subrutina BUFFER.

- Subrutina DISPLAY: saca por puerto D mediante direccionamiento indirecto los bytes almacenados entre las posiciones ECH y ADH (se va decrementando el apuntador). Cada vez que saca un contenido de memoria

activa la columna correspondiente cuando llama a la subrutina ONDA. Luego de llegar a la posición ADH y visualizarla repite el ciclo 7 veces para lograr la retención de la imagen. En esta subrutina se encuesta la bandera PIR1, 0 para la actualización del reloj en tiempo real. Nótese que la distribución de ECH hasta ADH corresponde a las 64 columnas del display correspondiendo la primera columna a la posición ECH. A continuación en la Figura 3.10 se muestra el diagrama en bloques de esta subrutina.

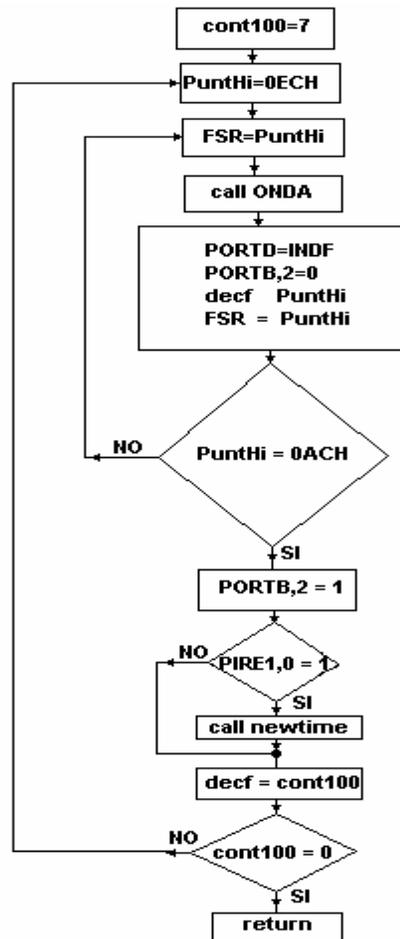


Figura 3.10. Subrutina de Display.

- Subrutina ROTAR: mediante direccionamiento indirecto va desplazando los valores almacenados en memoria de la posición siguiente a la posición actual direccionada por el apuntador. Este proceso comienza a partir de la posición ADH y se va incrementando el apuntador hasta llegar a la posición

EEH. Una vez que el apuntador llega a este valor se retorna de esta subrutina. Nótese que luego de rotar todas las posiciones en las dos posiciones mas altas del buffer se encuentra repetida la información pero esto no afecta ya que en el orden sucesivo primero se llama a BUFFER para que actualice la posición mas alta y luego a DISPLAY para que visualice. A continuación en la Figura 3.11 se muestra el diagrama en bloques de esta subrutina.

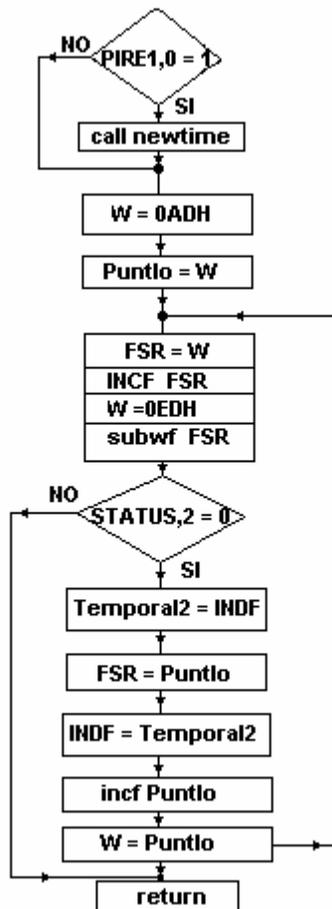


Figura 3.11. Subrutina ROTAR.

- Subrutina ONDA: genera una onda cuadrada utilizando el Timer0 que actúa como señal de reloj para los registros de desplazamiento. Dicha onda cuadrada tiene una frecuencia de 4.7 KHz, pero se varío el ciclo útil haciendo que el pulso en alto ocupe casi la totalidad del tiempo de la onda

cuadrada (99%). Esto se hace con el objetivo de que permanezca la columna activada el mayor tiempo posible lo cual permite que la intensidad de iluminación sea mayor. Para lograr esto cuando la onda cuadrada esta en '1' se carga el Timer0 con 150 lo que hace que cuente hasta el desborde (255) 105 pulsos y dado que se configuró el Timer con valor de 2 en el preescalador, el pulso en alto dura 210 μ s. Una vez que se desborda el Timer0 se pasa la onda cuadrada a '0' y se carga el Timer0 con 255 de manera que el pulso en bajo de la onda cuadrada dura solamente 2 μ s. A continuación en la Figura 3.12 se muestra el diagrama en bloques de esta subrutina.

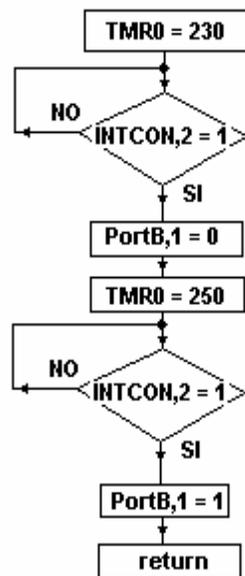


Figura 3.12. Subrutina ONDA.

- Subrutina LIMP: se encarga de limpiar las posiciones de memoria que conforman el buffer antes de que la subrutina Display las visualice, escribiendo '0' en cada una de ellas. Para ello carga en el registro PuntHi la dirección más alta del buffer y pone al FSR a apuntar a esta dirección. Luego utilizando direccionamiento indirecto escribe '0' en esa posición. Posteriormente se decrementa PuntHi y se le asigna al FSR. Cada vez que decrementa PuntHi se pregunta si ya llego a la posición mas baja. Si llego,

sale de la subrutina; si no, repite el ciclo escribiendo 0 en la posición de memoria que apunta el FSR. A continuación en la Figura 3.13 se muestra el diagrama en bloques de esta subrutina.

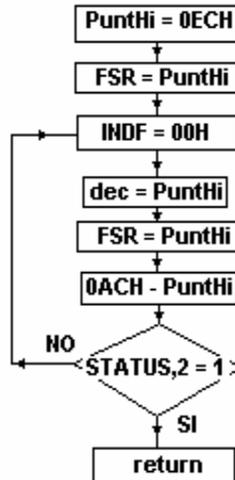


Figura 3.13. Subrutina LIMP.

- Subrutina Llenar: es la encargada de tomar el código ASCII de los caracteres del mensaje enviado por teclado o por comunicación serie y almacenarlos en memoria. Para esto tiene un espacio disponible de 60 bytes desde 20h hasta 5CH. Esta subrutina funciona recibiendo en dato el ASCII a guardar, luego apunta con el FSR a la dirección que tiene dircom y utilizando direccionamiento indirecto escribe en esta posición el ASCII recibido. Una vez almacenado el ASCII se pregunta si se llegó a la posición 5CH o si el carácter recibido fue el ASCII del punto '.'. Si esto ocurre se reinician los punteros y se sale de la subrutina; si no, se incrementa dircom y se sale de la subrutina. A continuación en la Figura 3.14 se muestra el diagrama en bloques de esta subrutina.

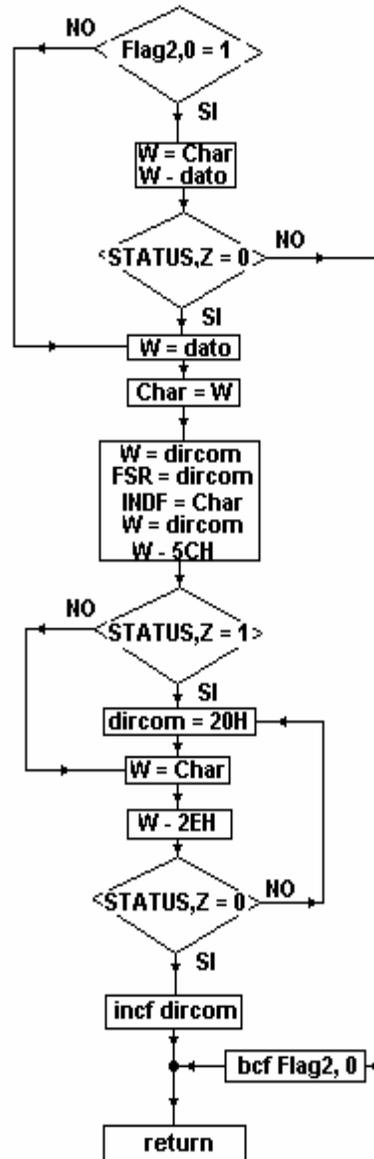


Figura 3.14. Subrutina Llenar.

- Subrutina Sacar: se encarga de extraer de memoria los códigos ASCII almacenados a partir de la posición 20H, mediante direccionamiento indirecto. Cada vez que se extrae el caracter se retorna de subrutina. Cuando el caracter extraído es el punto '.', o el apuntador llega a la ultima posición de memoria reservada para el almacenamiento de caracteres 5CH, dicho apuntador se reinicializa con su valor original. A continuación en la Figura 3.15 se muestra el diagrama en bloques de esta subrutina

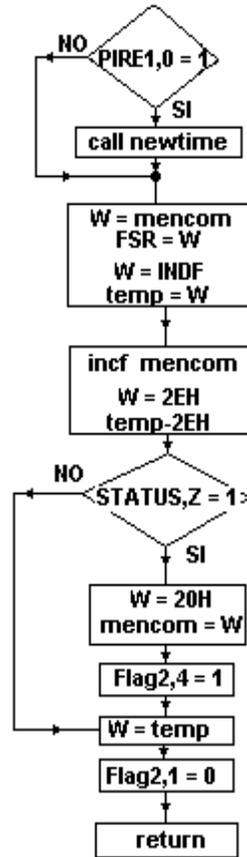


Figura 3.15. Subrutina Sacar.

- Reloj de tiempo real: para el reloj de tiempo real se utilizó el timer 1 que contiene 16 bits el cual se habilita una vez que el usuario actualiza la hora mediante la comunicación por teclado o utilizando el puerto serie. El timer se inicializa con el valor 0BDBh. Como el desborde ocurre al llegar a FFFF, en cada desborde el timer cuenta 62500 pulsos. Según la fórmula:

$$\text{TEMP} = (4/F_{\text{Osc}}) * \text{Prescalador} * \text{Timer} \quad (3.2)$$

Sustituyendo para las variables de nuestra aplicación:

$$F_{\text{Osc}} = 4 \text{ MHz} \quad (\text{valor de frecuencia del cristal})$$

$$\text{Prescalador} = 8 \quad (\text{configuración del timer 1})$$

Tenemos:

TEMP= 0.5s

Por tanto para implementar la case de tiempo de 1 minuto necesaria para actualizar el reloj de tiempo real se necesitan contar $60/0.5= 120$ desbordes del timer 1. El timer 1 se configura por interrupción la cual se genera por cada desborde del mismo. La subrutina de atención a la interrupción se encarga de contar los desbordes y una vez obtenidos 120 desbordes se incrementa la variable MINUTO y se comprueba si dicha variable llegó al valor 60. Cuando esto ocurre se limpia esta variable y se incrementa la variable HORA y se comprueba si esta llegó al valor 24. Una vez que ocurre esto, se limpia la variable HORA. Para visualizar el valor de la hora es necesario procesar el mismo. El procesamiento se realiza dentro del lazo de visualización y consiste en dividir los valores de las variables MINUTO y HORA entre 10. El valor de los cocientes obtenidos son los valores de las decenas de minutos y hora respectivamente, mientras que los valores de resto obtenidos son los valores de las unidades de minuto y hora respectivamente.

- Programa principal: el programa principal está formado por los siguientes bloques:
 1. Bloque de inicialización: en este bloque se llaman las subrutinas que se encargan de inicializar las variables que se utilizan en toda la aplicación, así como la configuración y habilitación de las interfaces utilizadas.
 2. Lazo loop: este lazo se encarga de almacenar en memoria a partir de la posición los códigos ASCII de los caracteres que conforman el mensaje de bienvenida predefinido.
 3. Lazo loop2: este es el lazo principal de la aplicación, pues es el responsable de sacar de memoria los códigos ASCII de mensaje, la

hora y la temperatura y visualizarlos; además en este lazo se realiza el refrescamiento del cartel.

4. Lazo loop3: en este lazo se introduce en memoria el mensaje introducido por teclado.
5. Lazo loop31: en este lazo se actualiza la hora mediante el teclado.
6. Lazo loopser: en este lazo se introduce en memoria un cartel utilizando la comunicación por puerto serie.
7. Lazo serhour: en este lazo se actualiza la hora a través de la comunicación por puerto serie.

El diagrama de bloques del programa principal se muestra en la Figura 3.16.

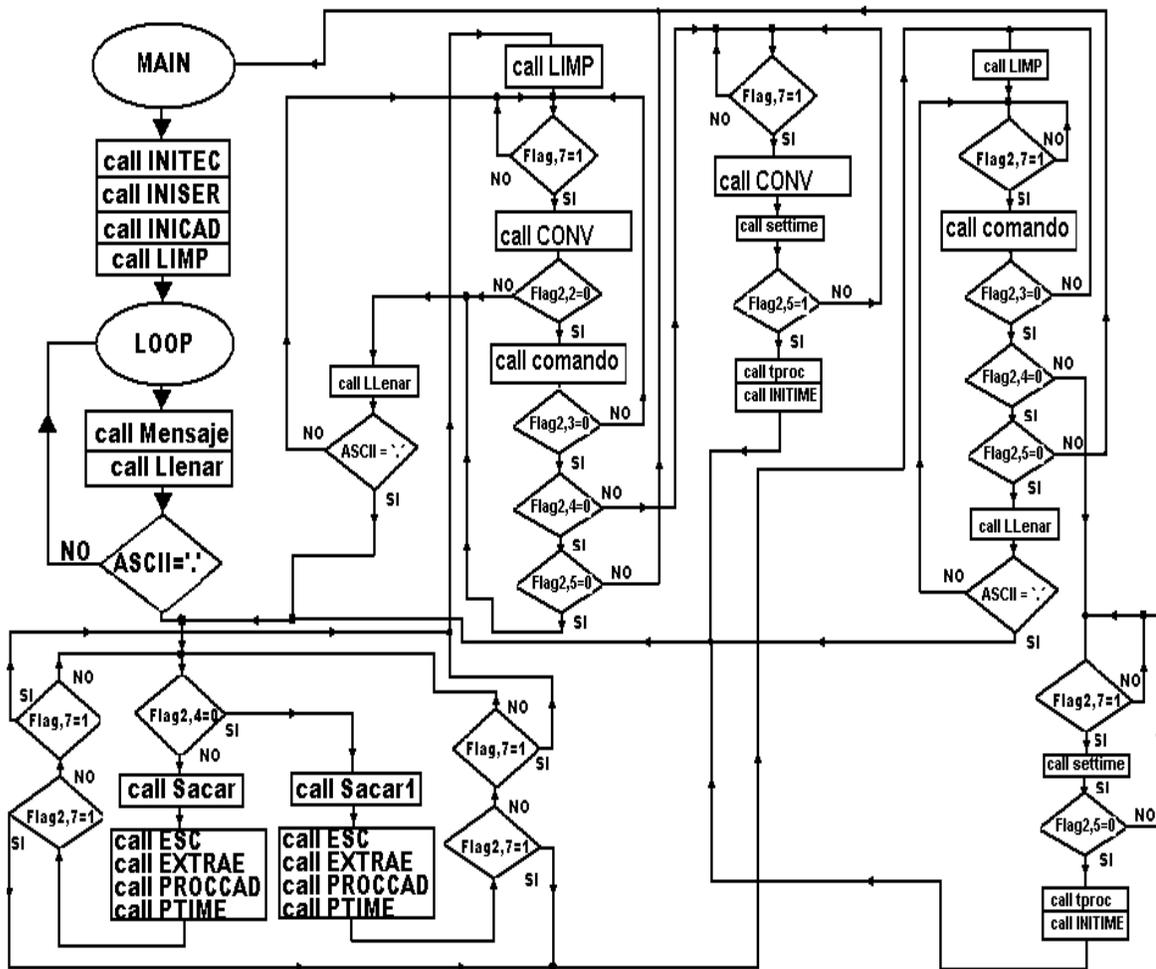


Figura 3.16. Diagrama de bloques del programa principal.

3.3.2 Comunicación PC-cartel pasa-mensajes.

La comunicación serie PC-PIC se realiza a través del puerto serie. Los caracteres que conforman el mensaje o los caracteres para la actualización de la hora son enviados a través de un programa emulador de terminal. En este caso se ha escogido al Hyper Terminal porque es una versión muy popular ya que viene incluido en la instalación de Microsoft Windows en cualquiera de sus versiones.

Al 'Hyper Terminal' se llega a través del menú 'INICIO'; luego 'Programas' ; después 'Accesorios'; más tarde 'Comunicaciones' y finalmente 'Hyper Terminal' donde aparecerá una nueva conexión como se muestra en la Figura 3.17.



Figura 3.17. Ventana de inicio de conexión en el Hyper Terminal.

A partir de aquí se le da un nombre a la conexión y se escoge un icono para que la represente. Después se selecciona el puerto de salida teniendo en cuenta la conexión física del cable. Esta conexión física del cable no necesita usar las líneas de control del puerto serie por lo que de las 9 vías del mismo solo se usarán los pines 2, 3 y 5 que representan 'recepción de datos', 'transmisión' y 'tierra del sistema' respectivamente. Luego aparecerá en pantalla la ventana de configuración de la sesión de Hyper Terminal como se muestra en la Figura 3.18.



Figura 3.18. Ventana de configuración de la sesión de Hyper Terminal.

Esta ventana de configuración está estrechamente relacionada con las características que posee el interfaz de comunicación serie (SCI) del PIC 16F877A para esta aplicación pues ambos casos los parámetros de configuración deben ser los mismos. En este diseño se escogió para la transmisión y recepción una velocidad de 1200 baudios. Los datos tienen un formato de 8 bits y un bit de parada, obviándose al bit de paridad. Con estos elementos se pueden llenar los 4 primeros parámetros de la Figura 3.18. El último (control de flujo) utiliza la opción 'Ninguno'.

Una vez configurada nuestra sesión ya se puede comenzar la comunicación con el PIC siguiendo los pasos ofrecidos en el Manual de Usuario, presentado en el epígrafe 3.5.

3.4 Análisis económico.

Haciendo un análisis de los precios de todas las componentes utilizadas en la construcción del cartel pasa-mensajes, se llegó a la conclusión de que el costo total del equipo es de aproximadamente 80.00 CUC. En el mercado un cartel pasa-mensajes con características similares cuesta alrededor de los 350 USD. El tipo de interfaz de comunicación utilizado encarece mucho más el precio, llegando

a valer algunos hasta los 2000 USD dependiendo de las características que ofrece cada producto en particular. (Mercado Libre, 2007)

El cartel diseñado posee facilidades de comunicación serie y por teclado PS/2; contiene además un reloj de tiempo de real y muestra el valor de la temperatura ambiente, prestaciones que lo harían más caro en caso de su adquisición en el mercado mundial. De esta forma se demuestra el gran aporte económico que este representa para la Facultad de Ingeniería Eléctrica.

3.5 Manual de usuario.

El cartel pasa-mensajes muestra una secuencia formada por un mensaje, la hora actual y la temperatura. Inicialmente, al ser energizado se muestra un mensaje predefinido almacenado en memoria: "BIENVENIDOS FIE UCLV", a continuación muestra la hora que se inicializa en 00:00 y luego la temperatura ambiente en el formato XX°C. El cartel en funcionamiento se muestra en el Anexo VII.

El usuario dispone de dos vías para actualizar el mensaje y la hora: comunicación por teclado PS/2 y comunicación a través del puerto serie para lo cual se utiliza el Hyper Terminal. En cualquiera de los dos casos la secuencia de pasos a seguir es la misma, solo hay que tener en cuenta que a la hora de utilizar la comunicación serie se deben escribir los caracteres en mayúscula.

Secuencia de escritura de mensaje:

1. Pulse una tecla para borrar el cartel.
2. Pulse el comando M, el cual indica que se va a escribir un mensaje.
3. Teclee el mensaje, el cual no debe exceder los 60 caracteres.
4. Teclee el punto para indicar el fin de mensaje.

Secuencia de actualización de la hora:

1. Pulse una tecla para borrar el cartel.
2. Pulse el comando H, el cual indica que se va a actualizar la hora.

3. Teclee la hora en formato militar utilizando 4 dígitos, los dos primeros para la hora, y los últimos para los minutos. Ejemplo: 1300 (1:00pm).
4. Teclee el punto para indicar el fin de escritura.

Si luego de haber borrado el cartel se pulsa una tecla que no sea ninguno de los comandos, el cartel visualizará nuevamente el mensaje de bienvenida predefinido. Igualmente un error durante la actualización de la hora, provoca que se visualice el mensaje de bienvenida predefinido.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Se realizó un estudio profundo de algunos de los subsistemas avanzados más importantes presentes en los microcontroladores PICs.
2. Se lograron realizar aplicaciones sencillas con estos subsistemas, pero que constituyen un gran apoyo para la docencia en particular a las asignaturas Microprocesadores I y II.
3. Se integraron estos subsistemas en una aplicación de mayor complejidad y también con gran valor docente.
4. Se obtuvo como resultado de la aplicación un cartel pasa-mensajes para la Facultad de Ingeniería Eléctrica, el cual posee un gran valor social como elemento decorativo e informativo para la misma.

Recomendaciones

1. Continuar el estudio de otros subsistemas avanzados de los PICs, particularmente la comunicación por puerto USB y emplear estos subsistemas en nuevas aplicaciones
2. Tomar como base este cartel pasa-mensajes y perfeccionar sus funciones, añadiéndole más prestaciones.
3. Utilizar software de alto nivel para implementar interfaces de comunicación con el cartel pasa-mensajes.

REFERENCIAS BIBLIOGRÁFICAS

- Angulo, I. y J. M. Angulo, (1999) *Microcontroladores PIC. Diseño práctico de aplicaciones*. Edición de Mc Graw Hill. Madrid, Interamericana de España.
- Bazán, C., (2006) "La familia de los PICs". *Conferencias del plan de estudio, asignatura Microcontroladores II*. 2006, Santa Clara, Facultad de Ingeniería Eléctrica, Universidad Central "Marta Abreu" de Las Villas.
- Chapweske, A. (2003). "The PS2 mouse/keyboard Protocol". [En línea]. Disponible en: <http://www.computer-engineering.org/ps2protocol> [Consultado el 4 de mayo de 2007]
- Fairchild Semiconductor. (2003). "KA78XXE/KA78XXA E3-Terminal 1A Positive Voltage Regulator". [En línea]. Fairchild Semiconductor, disponible en: <http://pdf1.alldatasheet.com/datasheet-pdf/view/116342/FAIRCHILD/KA7805AE.html> [Consultado el 18 de mayo de 2007]
- Fairchild Semiconductor. (2000). "DM74LS164 8-Bit Serial In/Parallel Out Shift Register". [En línea]. Fairchild Semiconductor, disponible en: http://www.datasheet4u.com/html/7/4/L/74LS164_FairchildSemiconductor.pdf.html [Consultado el 23 de mayo de 2007]
- Herrera, D. y A. Hernández. (2004). "Descripción del PIC 16F877A". [En línea]. Maracay, disponible en: <http://www.monografias.com/trabajos18/descripcion-pic/descripcion-pic.shtml> [Consultado el 10 de mayo de 2007]
- Indicart Carteles Electrónicos. (2001). "Letreros pasa-mensajes". [En línea]. Disponible en: <http://www.indicart.com.ar/pasamens.htm> [Consultado el 4 de mayo de 2007]
- López, D. (2005). "Diseño de rótulo luminoso con fines docentes". [En línea]. Disponible en: http://www.ciclope.info/display/matrix_es.shtml [Consultado el 1 de junio de 2007]
- Mercado Libre. (2007). "Cartel electrónico de texto pasante o pasa-mensaje". [En línea]. Disponible en: <http://articulo.mercadolibre.com.ar/MLA-26593646-cartel-electronico-de-texto-pasante-o-pasamensajes-JM> [Consultado el 26 de marzo de 2007]

- Microchip. (1997). "PICmicro™ Mid-Range MCU Family Reference Manual". [En línea]. Microchip, disponible en: <http://ww1.microchip.com/downloads/en/devicedoc/33023a.pdf> [Consultado el 6 de noviembre de 2006]
- Microchip. (2003a). "MPLAB IDE v6.xx Quick Start Guide". [En línea]. Microchip, Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/51519a.pdf> [Consultado el 18 de enero de 2007]
- Microchip. (2003b). "PIC16F87XA Data Sheet: 28/40/44-Pin Enhanced Flash Microcontrollers". [En línea]. Microchip, disponible en: <http://picmania.garcia-cuervo.com/recursos/PIC%2016F87xA.pdf> [Consultado el 7 de noviembre de 2006]
- Morell, R. et al., (2007). "Atención a teclado PS2 con PIC." [En línea]. Santa Clara, disponible en: \\neumann\Asignaturas\Telecom y Electrónica\Microprocesadores2\CURSO 06-07\tarea extraclase\tareas\teclado [Consultado el 29 de marzo de 2007]
- National Semiconductor. (1997). "LM35/LM35A/LM35C/LM35CA/LM35D: Precision Centigrade Temperature Sensors". [En línea]. National Semiconductor, disponible en: <http://picmania.garcia-cuervo.com/recursos/lm35a.pdf> [Consultado el 6 de noviembre de 2006]
- Palacherla, A. (1997). "Software Implementation of Asynchronous Serial I/O". [En línea]. Microchip, disponible en: <http://ww1.microchip.com/downloads/en/AppNotes/00555c.pdf> [Consultado el 6 de marzo de 2007]
- Pino, J. (2004). "Display Electrónico "Publik" con matriz de LEDs". [En línea]. Disponible en: http://www.josepino.com/pic_projects/indexsp.pl?matrix_display.jpg [Consultado el 14 de marzo de 2007]
- Maxim Integrated Products. (2004). "+5V-Powered, Multichannel RS-232 Drivers/Receivers". [En línea]. Maxim Integrated Products, disponible en: <http://picmania.garcia-cuervo.com/recursos/MAX232CPE.pdf> [Consultado el 5 de noviembre de 2006]
- Sánchez, L. (2005). "Microcontroladores genéricos". [En línea]. Uruguay, Disponible en: <http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml> [Consultado el 25 de abril de 2007]
- Segredo, R., (2006) *Aplicaciones docentes para microcontroladores PICs*. Trabajo de Diploma. Santa Clara, Facultad de Ingeniería Eléctrica, Universidad Central "Marta Abreu" de las Villas.
- Tebus, R. et al., (2007). "Medición de temperatura con PIC". [En línea]. Santa Clara, Disponible en: \\neumann\Asignaturas\Telecom y

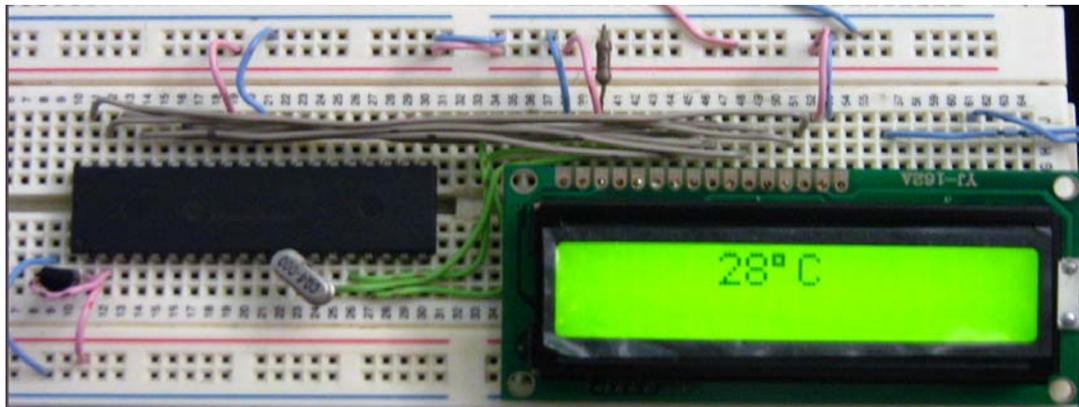
- Electronica\Microprocesadores2\CURSO 06-07\tarea
extraclase\tareas\proyecto CAD [Consultado el 6 de mayo de 2007]
- Valdés, Y. et al., (2007). "Transmisión Serie con 16F877A". [En línea]. Santa Clara, Disponible en: \\neumann\Asignaturas\Telecom y
Electronica\Microprocesadores2\CURSO06-07\tareaextraclase\tareas\serial
communication puerto [Consultado el 16 de marzo de 2007]
- Wikipedia. (2007). "Microcontroladores PIC". [En línea]. Disponible en:
[http://es.wikipedia.org/wiki/Microcontrolador PIC](http://es.wikipedia.org/wiki/Microcontrolador_PIC) [Consultado el 2 de junio
de 2007]

ANEXOS

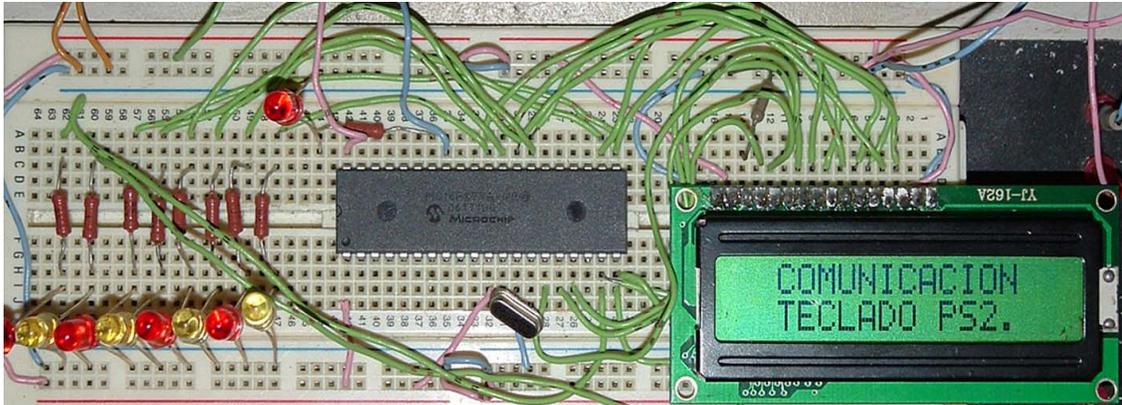
Anexo I. Comunicación serie PC-PIC con visualización a través de display LCD.



Anexo II. Medición de temperatura con visualización a través de display LCD.



Anexo III. Comunicación teclado-PIC con visualización a través del display LCD.



Anexo IV. Valores de SCANCODE para los caracteres de teclado.

KEY	MAKE	KEY	MAKE	KEY	MAKE
A	1C	9	46	[54
B	32	`	0E	INSERT	E0,70
C	21	-	4E	HOME	E0,6C
D	23	=	55	PG UP	E0,7D
E	24	\	5D	DELETE	E0,71
F	2B	BKSP	66	END	E0,69
G	34	SPACE	29	PG DN	E0,7A
H	33	TAB	0D	U ARROW	E0,75
I	43	CAPS	58	L ARROW	E0,6B
J	3B	L SHFT	12	D ARROW	E0,72
K	42	L CTRL	14	R ARROW	E0,74
L	4B	L GUI	E0,1F	NUM	77
M	3A	L ALT	11	KP /	E0,4A
N	31	R SHFT	59	KP *	7C
O	44	R CTRL	E0,14	KP -	7B
P	4D	R GUI	E0,27	KP +	79
Q	15	R ALT	E0,11	KP EN	E0,5A
R	2D	APPS	E0,2F	KP.	71
S	1B	ENTER	5A	KP 0	70
T	2C	ESC	76	KP 1	69
U	3C	F1	05	KP 2	72
V	2A	F2	06	KP 3	7A
W	1D	F3	04	KP 4	6B
X	22	F4	0C	KP 5	73
Y	35	F5	03	KP 6	74
Z	1A	F6	0B	KP 7	6C
0	45	F7	83	KP 8	75
1	16	F8	0A	KP 9	7D
2	1E	F9	01]	5B
3	26	F10	09	;	4C
4	25	F11	78	'	52
5	2E	F12	07	,	41
6	36	PRNT SCRN	E0,12, E0,7C	.	49
7	3D	SCROLL	7E	/	4A
8	3E	PAUSE	E1,14,77, E1,F0,14, F0,77		

Anexo V. Código ASCII para caracteres de teclado.

Caracteres imprimibles										
Dec	Hex	Car.		Dec	Hex	Car.		Dec	Hex	Car.
32	20	Espacio		64	40	@		96	60	`
33	21	!		65	41	A		97	61	a
34	22	"		66	42	B		98	62	b
35	23	#		67	43	C		99	63	c
36	24	\$		68	44	D		100	64	d
37	25	%		69	45	E		101	65	e
38	26	&		70	46	F		102	66	f
39	27	'		71	47	G		103	67	g
40	28	(72	48	H		104	68	h
41	29)		73	49	I		105	69	i
42	2A	*		74	4A	J		106	6A	j
43	2B	+		75	4B	K		107	6B	k
44	2C	,		76	4C	L		108	6C	l
45	2D	-		77	4D	M		109	6D	m
46	2E	.		78	4E	N		110	6E	n
47	2F	/		79	4F	O		111	6F	o
48	30	0		80	50	P		112	70	p
49	31	1		81	51	Q		113	71	q
50	32	2		82	52	R		114	72	r
51	33	3		83	53	S		115	73	s
52	34	4		84	54	T		116	74	t
53	35	5		85	55	U		117	75	u
54	36	6		86	56	V		118	76	v
55	37	7		87	57	W		119	77	w
56	38	8		88	58	X		120	78	x
57	39	9		89	59	Y		121	79	y
58	3A	:		90	5A	Z		122	7A	z
59	3B	;		91	5B	[123	7B	{
60	3C	<		92	5C	\		124	7C	
61	3D	=		93	5D]		125	7D	}
62	3E	>		94	5E	^		126	7E	~
63	3F	?		95	5F	_		127	7F	DEL

Anexo VI. Circuito impreso del cartel pasa-mensajes.



Anexo VII. El cartel pasa-mensajes.

